Microsoft System Center

Cloud Management with App Controller

Yung Chou ▪ Keith Mayer
Mitch Tulloch, Series Editor

Visit us today at

microsoftpressstore.com

- **Hundreds of titles available** – Books, eBooks, and online resources from industry experts
- **Free U.S. shipping**
- **eBooks in multiple formats** – Read on your computer, tablet, mobile device, or e-reader
- **Print & eBook Best Value Packs**
- **eBook Deal of the Week** – Save up to 60% on featured titles
- **Newsletter and special offers** – Be the first to hear about new releases, specials, and more
- **Register your book** – Get additional benefits

Microsoft

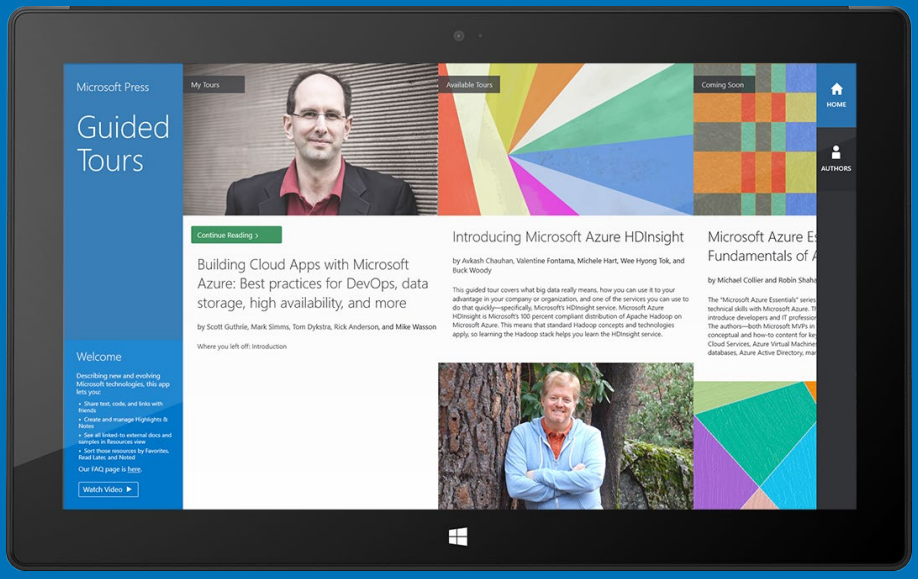
Get the latest news from Microsoft Press sent to your inbox.

- New and upcoming books
- Special offers
- Free eBooks
- How-to articles

Sign up today at MicrosoftPressStore.com/Newsletters

Microsoft

Wait, there's more...


Microsoft Press
Guided Tours
My Tours
Available Tours
Coming Soon
HOME
AUTHORS
Continue Reading
Building Cloud Apps with Microsoft Azure: Best practices for DevOps, data storage, high availability, and more
by Scott Guthrie, Mark Simms, Tom Dykstra, Rick Anderson, and Mike Wasson
Where you left off: Introduction
Introducing Microsoft Azure HDInsight
by Avkash Chauhan, Valentine Fontama, Michele Hart, Wee Hyong Tok, and Buck Woody
Welcome
Watch Video

Find more great content and resources in the **Microsoft Press Guided Tours** app.

The Microsoft Press Guided Tours app provides insightful tours by Microsoft Press authors of new and evolving Microsoft technologies.

- Share text, code, illustrations, videos, and links with peers and friends
- Create and manage highlights and notes
- View resources and download code samples
- Tag resources as favorites or to read later
- Watch explanatory videos
- Copy complete code listings and scripts

Download from Windows Store

From technical overviews to drilldowns on special topics, get *free* ebooks from Microsoft Press at:

www.microsoftvirtualacademy.com/ebooks

Download your free ebooks in PDF, EPUB, and/or Mobi for Kindle formats.

Look for other great resources at Microsoft Virtual Academy, where you can learn new skills and help advance your career with free Microsoft training delivered by experts.

Microsoft Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2013 Microsoft Corporation (All)

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2013952564
ISBN: 978-0-7356-8308-2

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at *http://www.microsoft.com/learning/booksurvey*.

Microsoft and the trademarks listed at *http://www.microsoft.com/about/legal/en/us/IntellectualProperty/*Trademarks/EN-US.aspx are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: Diane Kohnen, S4Carlisle Publishing Services
Copyeditor: Andrew Jones
Cover Illustration: Twist Creative • Seattle
Cover Design: Microsoft Press Brand Team

Contents

Introduction *vii*

Chapter 1 App Controller essentials **1**

System requirements 1

Installation prerequisites 1

Windows Assessment and Deployment Kit for Windows 8.1 2

Installation user and App Controller service account 3

Microsoft SQL Server instance 3

Performing the installation 4

Product key 5

Prerequisites checker 5

Installation path 6

App Controller services 7

SSL certificate 8

SQL Server instance and App Controller database 9

Reviewing the installation results 10

Verifying installation log files 10

Verifying App Controller services 11

Role-based security model 11

User roles and delegation 12

Fabric visibility 14

Operations model and UI 14

App Controller resource configuring 14

App Controller UI 15

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Chapter 2 Managing private clouds 17

Which private clouds can be managed? 17

App Controller and Virtual Machine Manager 18

Preparing for self-service private cloud management 19

Signing in at the portal 21

Branding the portal experience 22

Connecting to private clouds using App Controller 22

Adding a network file share to App Controller 26

Managing Run As accounts 28

Deploying new workloads to private clouds 30

Managing private cloud workloads 38

Moving files to/from private clouds 41

Chapter 3 Managing public clouds 43

Why public cloud? 44

Introducing Windows Azure 44

Managing Windows Azure with the Windows Azure Management Portal 45

Managing Windows Azure with System Center 2012 R2 App Controller 46

Preparing for self-service public cloud management 47

Creating a self-signed management certificate 47

Uploading a management certificate to Windows Azure 50

Connecting to public clouds 52

Delegating access to public clouds 55

Creating a Windows Azure storage account 58

Deploying new workloads to a public cloud 60
Configuring a cloud service 63
Configuring a virtual network 64
Configuring a virtual machine 65
Ready to deploy 69
Managing public cloud workloads. 70
Managing files, disks, and images in public clouds 72
Moving files to/from Windows Azure storage accounts 72
Adding disks and images 73

Chapter 4 Managing hybrid clouds 75

Copying a VHD from VMM to Windows Azure. 75
Deploying a cloud service in Windows Azure using an uploaded VHD 78
Destination 79
Container 79
Topology and logistics 81
Payload 81
Completing the deployment 85
Copying virtual machines from VMM to Windows Azure. 87

Chapter 5 App Controller cmdlets 93

How App Controller cmdlets work. 93
Why App Controller cmdlets? 94
Importing the AppController module 94
Connecting with the App Controller server 96
Connecting to VMM. 97
Connecting to Windows Azure. 98

Adding a library share to copy and paste resources between clouds 101

Adding a VHD to a Windows Azure storage account container 102

Adding a VHD to a Windows Azure image store 103

Acquiring a VHD from a virtual machine, template, or the VMM library. 104

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Introduction

Microsoft System Center 2012 R2 App Controller is uniquely positioned as both an enabler and a self-service vehicle for connecting clouds and implementing the hybrid computing model. In Microsoft's cloud computing solutions, both System Center and Windows Azure play critical roles. System Center can be used to transform enterprise IT from a device-based infrastructure and deployment strategy to a service-based user-centric consumption model based on private cloud computing. Windows Azure on the other hand is a subscription-based public cloud platform that enables the development, deployment, and management of cloud solutions. App Controller is the glue that unifies these two platforms by providing a single interface that enables administrators to perform complex operations without overwhelming them with the underlying technical complexities involved.

This book serves as an introduction to implementing and managing the hybrid computing solutions using App Controller. It describes the basic concepts, processes, and operations involved in connecting, consuming, and managing resources that are deployed both on and off premises. Each chapter provides a concise, self-contained walkthrough for a specific aspect of managing private, public, and hybrid clouds using App Controller.

While cloud computing is still evolving, the hybrid approach will likely continue to emerge as the go-to IT computing model for the foreseeable future. Using App Controller to strategically connect both on-premises System Center private clouds with off-premises deployments in both Windows Azure and third-party cloud hosting providers enables new scenarios, develops new possibilities, and offers exciting new opportunities that can help IT transition better into the next generation of enterprise computing.

Errata & book support

We've made every effort to ensure the accuracy of this content. Any errors that have been reported since this content was published are listed on our Microsoft Press site:

http://aka.ms/SCAppController/errata

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 1

App Controller essentials

Microsoft System Center 2012 R2 App Controller is a component of System Center, an extension of Virtual Machine Manager (VMM), and relatively easy to implement. App Controller is a web-based self-service vehicle to facilitate the deployment of virtual machines (VMs) and services. App Controller can connect to private clouds based on a VMM server, to Windows Azure subscriptions, and to third-party hosting providers and can manage resources among these three environments. Based on VMM's role-based security model which defines who can do what and to what extent, App Controller can delegate authority by modeling a business function as a user role, thereby noticeably simplifying the security administration and management of a multitenant environment. Above all, as hybrid cloud becomes an emerging platform for next-generation computing, App Controller enables deployment of such hybrid scenarios and helps accelerate their adoption.

This chapter covers some of the basics including the system requirements, prerequisites, installation, role-based security model, operations model, and user interface (UI) of App Controller.

System requirements

The system requirements for installing the App Controller server, the App Controller web console on a client computer, and the Windows PowerShell Module for App Controller can be found in the Microsoft TechNet Library at *http://technet.microsoft.com/library/dn249764.aspx* so they won't be repeated here. Note that an App Controller installation is an extension of a targeted VMM server which must be specified during installation of App Controller.

Installation prerequisites

This section summarizes the prerequisites for installing the App Controller server in your environment.

Windows Assessment and Deployment Kit for Windows 8.1

The Windows Assessment and Deployment Kit (ADK) for Windows 8.1 is a required component for installing System Center 2012 R2 App Controller. The ADK for Windows 8.1 is available as a free download from Microsoft at *http://www.microsoft.com/en-us/download/details.aspx?id=39306*. The ADK for Windows 8.1 is a realization of Microsoft deployment and assessment methodologies and includes a suite of free tools to facilitate and improve the quality of Windows deployment and fundamentally reduce the overall costs associated with deployment. The ADK for Windows 8.1 includes the following:

- **Application Compatibility Toolkit (ACT)** This can be used to build inventories and assess compatibility when migrating an application. The ACT uses a database instance that must be running on Microsoft SQL Server 2005 (or Express edition) or later.
- **Deployment Tools** These are tools can be used for customizing disk images and automating Windows deployments.
- **Windows Preinstallation Environment** Also known as Windows PE, this is a minimal operating system that can be used to prepare a computer for installation or servicing. Windows PE requires the Deployment Tools.
- **User State Migration Tool (USMT)** This can be used for migrating user data from an existing Windows installation to a new one. USMT includes three tools: ScanState, LocalState, and USMTUtils.
- **Volume Activation Management Tool (VAMT)** This can be used for automating and managing Windows activations of Windows and Microsoft Office. It employs a database which must be a Microsoft SQL Server 2008 (or Express edition) instance or later.
- **Windows Performance Toolkit (WPT)** This can be used to monitor and profile Windows operating systems and applications. WPT includes the Windows Performance Recorder, Windows Performance Analyzer, and Xperf tools.
- **Windows Assessment Toolkit** This is a 2.4 GB download that can be used to produce diagnostics and remediation information of a local system by running jobs to measure and record the performance, reliability, and functionality. The Windows Assessment Toolkit requires the Deployment Tools, Windows PE, WPT, and SQL Server 2012 Express which is also included in the download.

For installing App Controller, the Deployment Tools and Windows PE are especially essential. Figure 1-1 shows the initial installation screen for installing the ADK for Windows 8.1, which is currently in preview at the time of this writing.

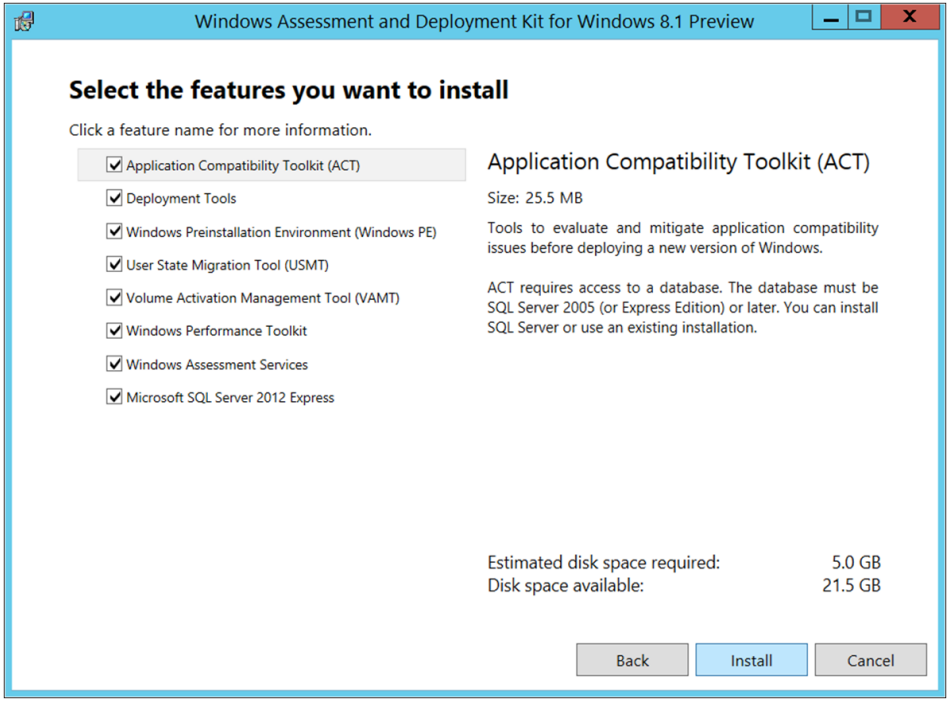

Windows Assessment and Deployment Kit for Windows 8.1 Preview
Select the features you want to install
Click a feature name for more information.
Application Compatibility Toolkit (ACT)
Deployment Tools
Windows Preinstallation Environment (Windows PE)
User State Migration Tool (USMT)
Volume Activation Management Tool (VAMT)
Windows Performance Toolkit
Windows Assessment Services
Microsoft SQL Server 2012 Express
Application Compatibility Toolkit (ACT)
Size: 25.5 MB
Tools to evaluate and mitigate application compatibility issues before deploying a new version of Windows.
ACT requires access to a database. The database must be SQL Server 2005 (or Express Edition) or later. You can install SQL Server or use an existing installation.
Estimated disk space required: 5.0 GB
Disk space available: 21.5 GB
Back
Install
Cancel

FIGURE 1-1 The components list for the Windows ADK.

At the end of the ADK installation, there is a check box to bring up the *ADK Getting Started Guide* which offers an overview of the ADK along with scenarios to help you better understand Microsoft's deployment and assessment methodologies. The guide now has a tile that can be pinned for frequent access as shown in Figure 1-2.

Installation user and App Controller service account

Installing App Controller on a server requires a domain user account with local Administrator privileges. The service account to run App Controller services can be the built-in Network Service account or a domain account.

Microsoft SQL Server instance

Prior to installing App Controller, be sure to identify a supported version of a Microsoft SQL Server instance in your environment or create a new instance. The user account installing App Controller must have at least database owner (DBO) permissions on the database associated with your App Controller installation.

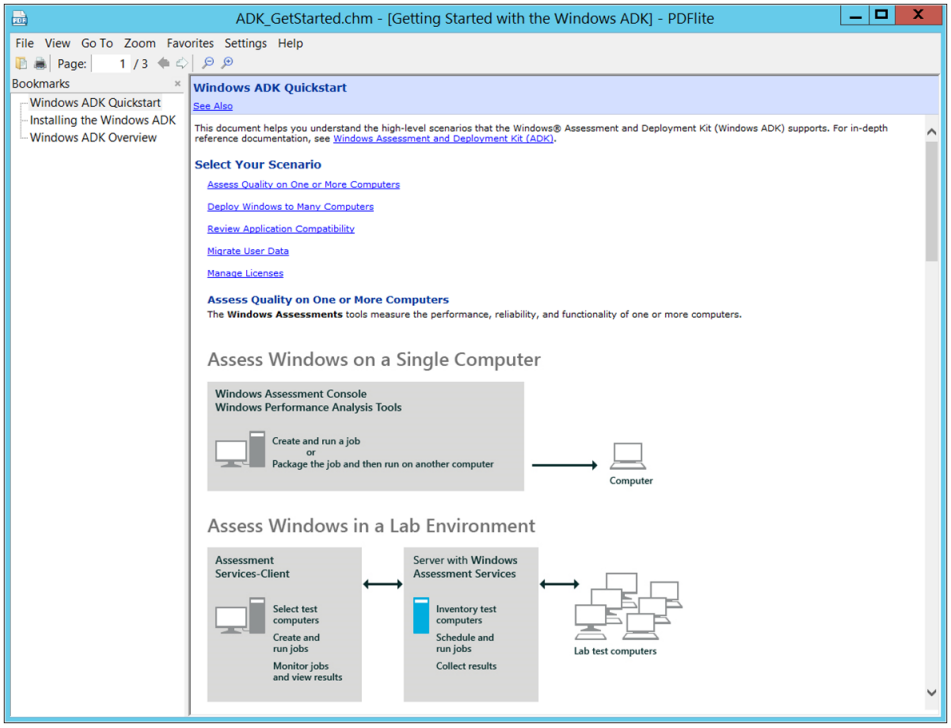

ADK_GetStarted.chm - [Getting Started with the Windows ADK] - PDFlite
File View Go To Zoom Favorites Settings Help
Page: 1 / 3
Bookmarks
Windows ADK Quickstart
Installing the Windows ADK
Windows ADK Overview
Windows ADK Quickstart
See Also
This document helps you understand the high-level scenarios that the Windows® Assessment and Deployment Kit (Windows ADK) supports. For in-depth reference documentation, see Windows Assessment and Deployment Kit (ADK).
Select Your Scenario
Assess Quality on One or More Computers
Deploy Windows to Many Computers
Review Application Compatibility
Migrate User Data
Manage Licenses
Assess Quality on One or More Computers
The Windows Assessments tools measure the performance, reliability, and functionality of one or more computers.
Assess Windows on a Single Computer
Windows Assessment Console
Windows Performance Analysis Tools
Create and run a job
or
Package the job and then run on another computer
Computer
Assess Windows in a Lab Environment
Assessment Services-Client
Select test computers
Create and run jobs
Monitor jobs and view results
Server with Windows Assessment Services
Inventory test computers
Schedule and run jobs
Collect results
Lab test computers

FIGURE 1-2 A view of the ADK Getting Started Guide.

Performing the installation

The System Center 2012 R2 App Controller installation process is very similar to that of System Center 2012 App Controller and is initialized by running Setup.exe as an administrator. The installation startup screen has links to important online content including the Release Notes, Installation Guide, and so on (see Figure 1-3). There is also an option on this screen to install the Windows PowerShell module for App Controller.

App Controller Setup
Microsoft System Center 2012 R2
App Controller
Install
You might have to restart the computer after you install App Controller.
Use Microsoft Update to check for updates to App Controller Setup.
Read the Update Services privacy statement
Standalone components
Install standalone components.
Install Windows PowerShell module for App Controller
Release Notes
Installation Guide
App Controller Online
System Center Online
Preview
© Copyright 2013 Microsoft Corporation. All rights reserved.
Close

FIGURE 1-3 The installation screen for App Controller Setup.

Product key

If you do not provide a product key during installation, App Controller will be installed as an evaluation edition. To provide a product key afterwards, simply rerun the setup program again and select the Upgrade option.

Prerequisites checker

There are a number of prerequisites for installing App Controller in an environment. When starting the installation process, a built-in prerequisites checker will identify the hardware/software components in place and suggest follow-up actions, as applicable, for any missing components. For example, Figure 1-4 shows an example of a blocked installation attempt where some prerequisites are missing. If desired, you can install the missing prerequisites at this time and then afterwards click the Verify Prerequisites Again link to rerun the prerequisites checker.

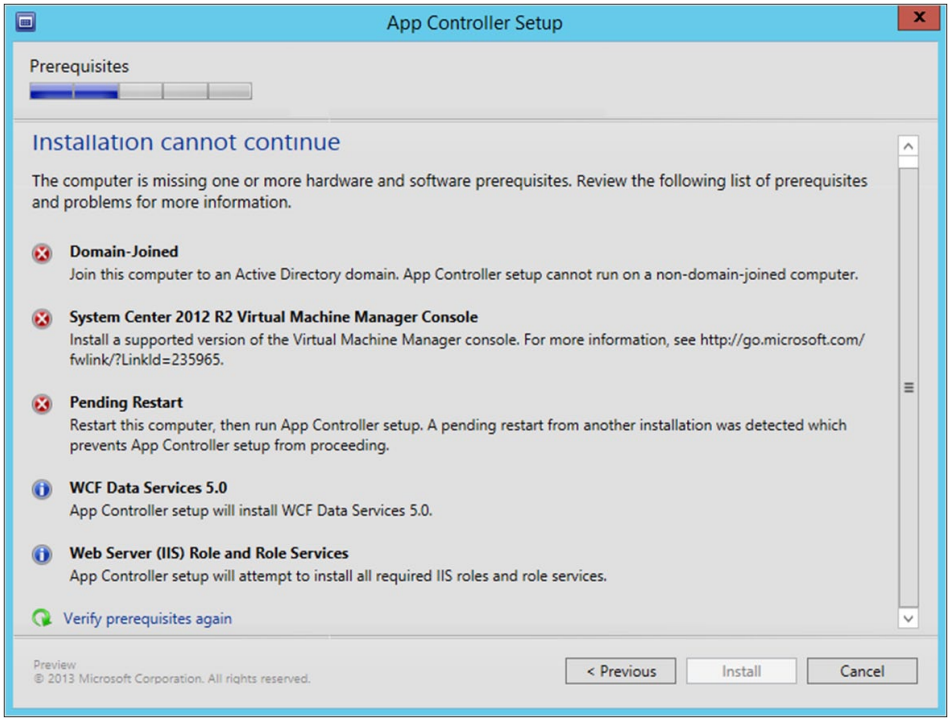

App Controller Setup
Prerequisites
Installation cannot continue
The computer is missing one or more hardware and software prerequisites. Review the following list of prerequisites and problems for more information.
Domain-Joined
Join this computer to an Active Directory domain. App Controller setup cannot run on a non-domain-joined computer.
System Center 2012 R2 Virtual Machine Manager Console
Install a supported version of the Virtual Machine Manager console. For more information, see http://go.microsoft.com/fwlink/?LinkId=235965.
Pending Restart
Restart this computer, then run App Controller setup. A pending restart from another installation was detected which prevents App Controller setup from proceeding.
WCF Data Services 5.0
App Controller setup will install WCF Data Services 5.0.
Web Server (IIS) Role and Role Services
App Controller setup will attempt to install all required IIS roles and role services.
Verify prerequisites again
Preview
© 2013 Microsoft Corporation. All rights reserved.
< Previous
Install
Cancel

FIGURE 1-4 The App Controller prerequisites checker indicating the installation has failed.

Once all of the prerequisites have been met, the Setup Wizard will continue and the installation process can proceed to the next step.

Installation path

By default, the setup program installs App Controller at C:\Program Files\Microsoft System Center 2012 R2\App Controller.

App Controller services

Either the built-in Network Service account or a domain account can be used as the service account for running the App Controller services. The default port for the internal communication of App Controller services is 18622 but this is customizable as shown in Figure 1-5.

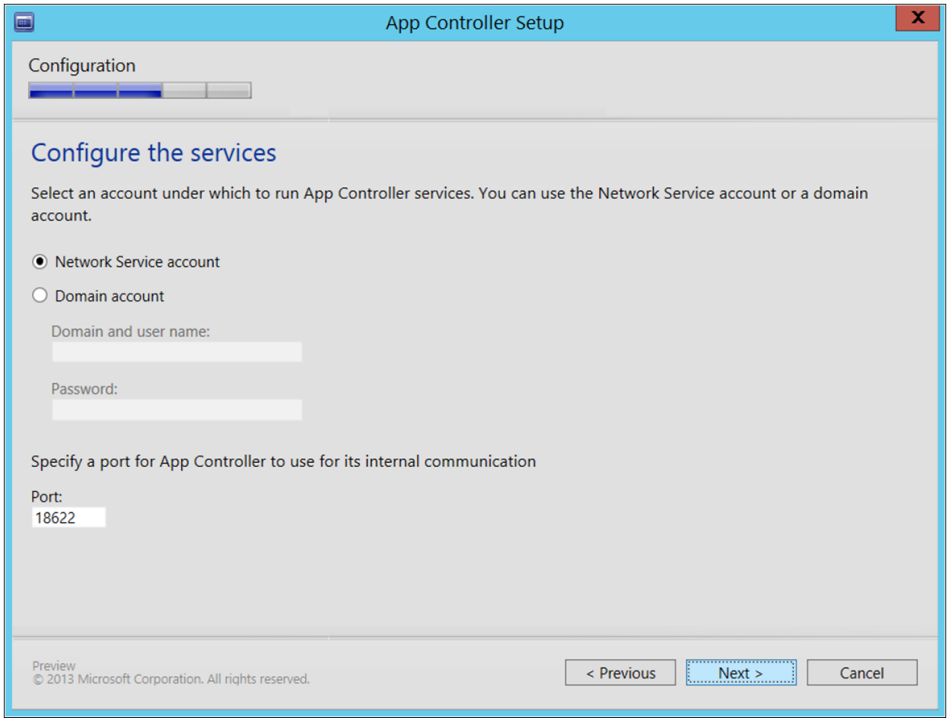

App Controller Setup
Configuration
Configure the services
Select an account under which to run App Controller services. You can use the Network Service account or a domain account.
Network Service account
Domain account
Domain and user name:
Password:
Specify a port for App Controller to use for its internal communication
Port:
18622
Preview
© 2013 Microsoft Corporation. All rights reserved.
< Previous
Next >
Cancel

FIGURE 1-5 You can configure the service account and port used by App Controller.

SSL certificate

The installation process provides the opportunity to specify the IIS website binding (IP address and TCP port). The default port is the SSL port 443 as shown in Figure 1-6. Setup can generate a self-signed certificate or you can select an existing x.509 certificate that has already been installed on the local machine. The figure shows an existing certificate named as ac.contoso.corp being designated as the SSL certificate for the App Controller website. By using IIS, which is required when installing App Controller, you can easily generate an SSL certificate using your enterprise public key infrastructure (PKI).

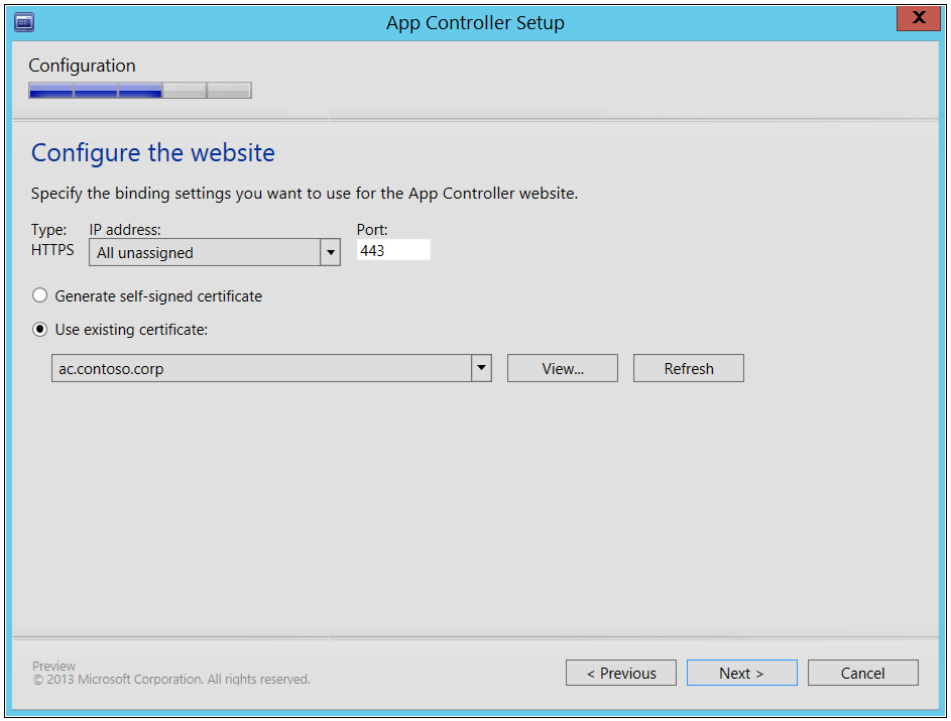


FIGURE 1-6 You can configure the IIS bindings and SSL certificate.

SQL Server instance and App Controller database

The default App Controller database is named AppController, as shown in Figure 1-7, but this is customizable.

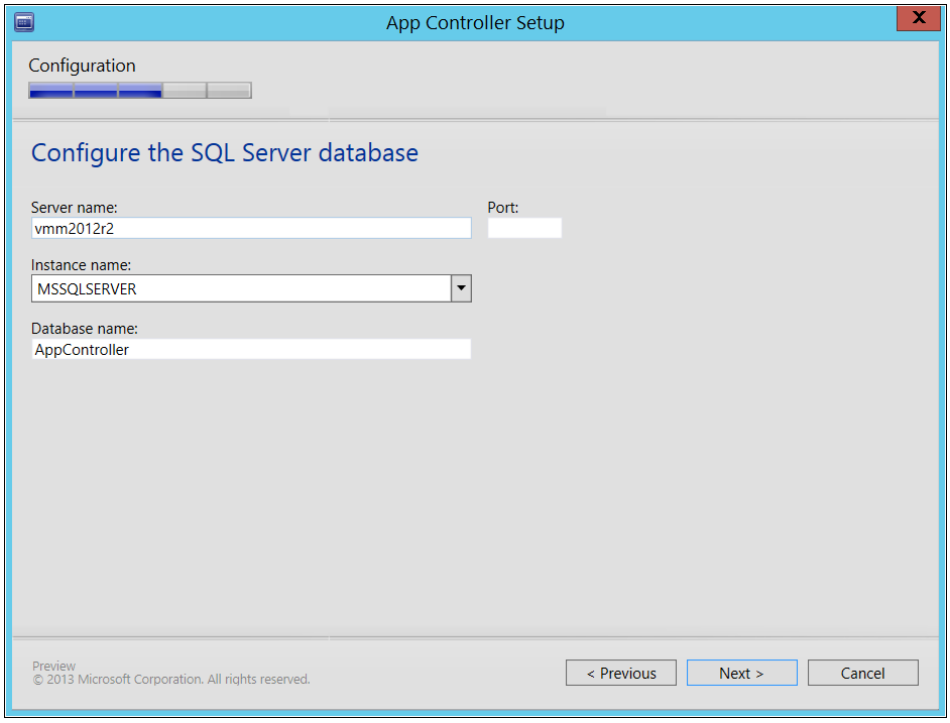


FIGURE 1-7 You can configure the SQL Server database for App Controller.

Reviewing the installation results

Once App Controller has been successfully installed as indicated by all checkmarks in a green circle on the final page of the App Controller Setup Wizard, be sure to review and document the installation logs by clicking the View Logs link as shown in Figure 1-8.

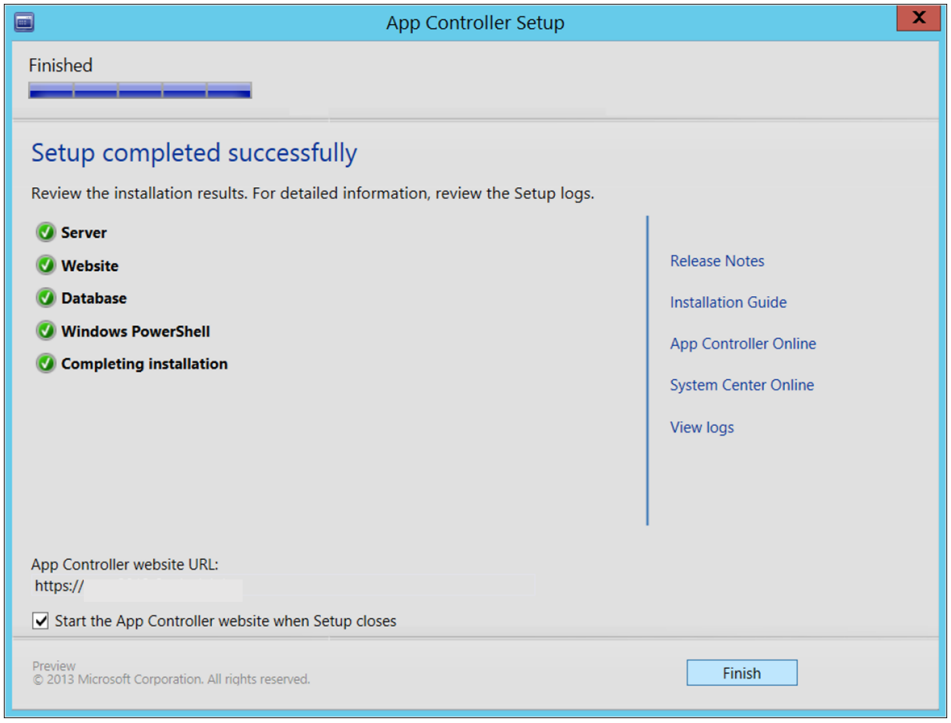


FIGURE 1-8 A view of the App Controller setup after a successful completion.

Verifying installation log files

The App Controller installation log files are stored in either %LOCALAPPDATA%\AppController\Logs or \ProgramData\AppControllerLogs. Figure 1-9 shows the log files folder of a typical App Controller installation.

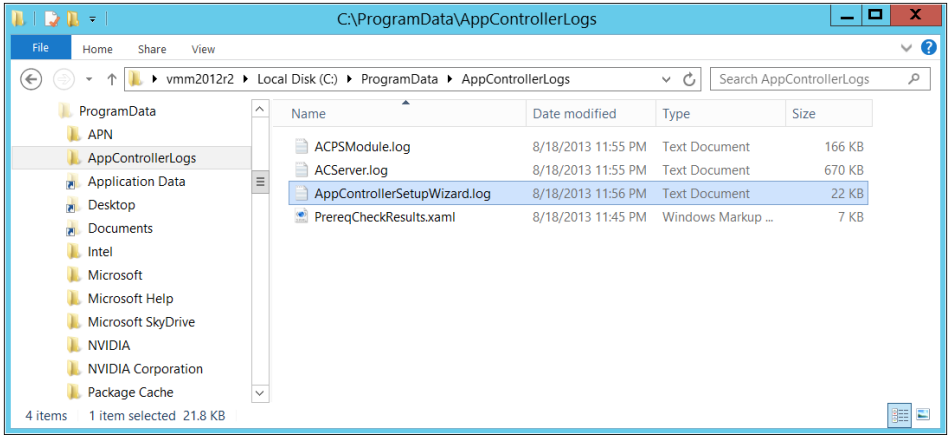


FIGURE 1-9 A view of the App Controller installation log files.

Verifying App Controller services

The Services node in Computer Management in Figure 1-10 shows that four services are installed by the App Controller setup program.

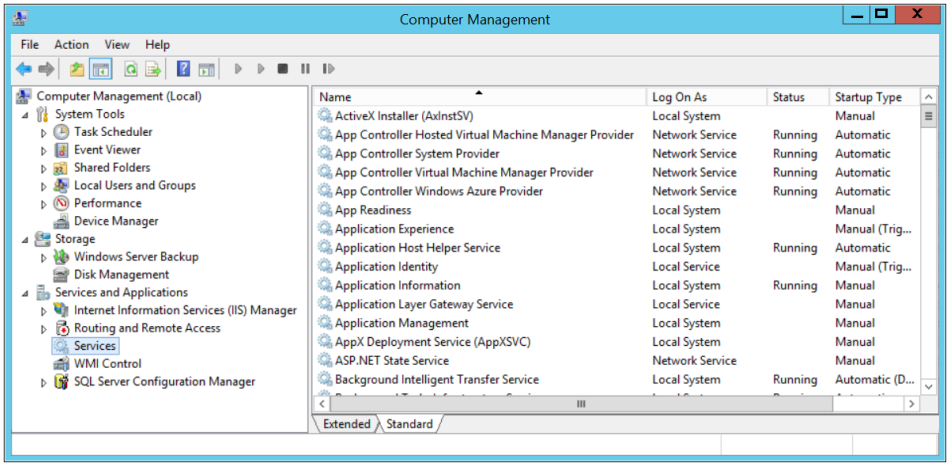


FIGURE 1-10 A view of the installed App Controller services.

Role-based security model

Before examining the experience of using App Controller, we will first review the App Controller security model to better understand the targeted usage scenarios. As mentioned earlier, App Controller is a self-service portal for an authorized user to manage service

deployments. The authorization model that App Controller uses is inherited from that of the associated VMM server. In the VMM administration console, the Security node in the Setting workspace can be used to define new user roles as shown in Figure 1-11.

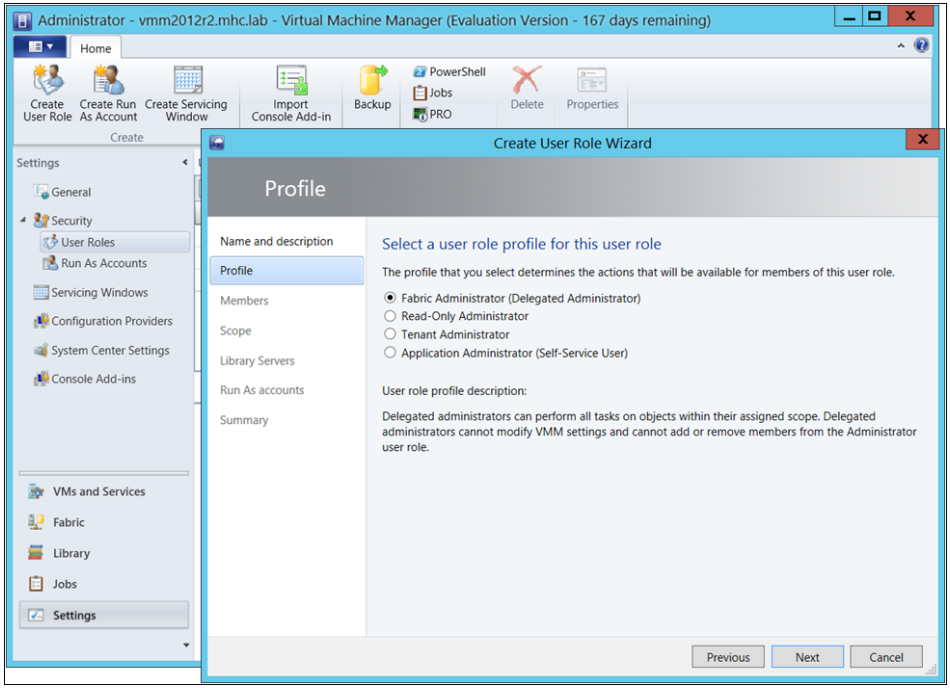


FIGURE 1-11 You can create and manage user roles using VMM.

User roles and delegation

A user role is a policy consisting of membership and a profile. The latter specifies a set of operations that can be operated on authorized objects. Specifically, a user role defines not only what tasks a user can perform on authorized resources, but also to what extent with what privileges such tasks can be performed. Once a user has been authenticated, those roles that the user is a member of are in effect.

A key benefit of this model is that with membership and a profile, that is, who and what to do, you can model an individual performing a specific business function with a particular set of tasks. This delegation model is called role-based security and significantly simplifies security administration because instead of specifying many individual operations on many individual objects, you can tie business functions to membership in a particular user role. By adding or removing a user from a user role, the user automatically inherits or is deprived of the operations, scopes, and privileges defined in the associated profile. Employing user roles also offers consistency in authorizing resources and provides a user-defined abstraction that translates security and administration requirements into the customer's business functions.

In System Center 2012 R2 App Controller there are four user role profiles. These roles are briefly described in the sections that follow.

Fabric Administrator (Delegated Administrator)

The Fabric Administrator role is a privileged role that can perform all tasks on authorized objects.

Read-Only Administrator

The Read-Only Administrator role can read the information of, but not modify, an object. The Read-Only Administrator role is intended for monitoring and auditing purposes.

Tenant Administrator

The Tenant Administrator role is a project/release/function leadership role. Users assigned this role can manage self-service users, virtual machines, and service deployment including user access and quotas.

Application Administrator (Self-Service User)

The Application Administrator role manages resources deployed by the individual. Users assigned this role can perform only those tasks specifically marked in the Permissions page of the profile. Figure 1-12 shows the list of tasks available for the Application Administrator role.

Name	Description
Author	Author virtual machine and service templates
Checkpoint	Create and manage virtual machine checkpoints
Checkpoint (Restore only)	Restore to but cannot create virtual machine checkpoints
Deploy	Create virtual machines and services from VHDs or templates
Deploy (From template only)	Create virtual machines and services from templates only
Local Administrator	Grants local administrator rights on virtual machines
Pause and resume	Pause and resume virtual machines and services
Receive	Receive resources from other self-service users
Remote connection	Remotely connect to virtual machines
Remove	Remove virtual machines and services
Save	Save virtual machines and services
Share	Share resources with other self-service users
Shut down	Shut down virtual machines
Start	Start virtual machines and services
Stop	Stop virtual machines and services
Store and re-deploy	Store virtual machines in the library, and re-deploy those virtual machines

FIGURE 1-12 A list of tasks available for the Application Administrator role.

Fabric visibility

Each of the above user roles can access resources using either the App Controller web-based interface or the VMM administration console. The visibility of the underlying fabric (that is, the servers, networking, and storage resource pools) will vary depending on user role. One key distinction of accessing resources with App Controller and VMM Admin Console is that App Controller does not reveal fabric regardless of whether the account is a VMM administrator or one with a Fabric Administrator role. However, accessing with VMM admin console, a VMM administrator and a Fabric Administrator will see fabric workspace while a Tenant Administrator or an Application Administrator will not. In fact, an idea of App Controller is to enable a service owner or technical leadership to manage a service deployment without concerning the underlying infrastructure and technical complexities. Limiting fabric visibility is here an advantage. For those who need an access to fabric, log in a VMM admin console instance instead.

Operations model and UI

This section briefly describes the App Controller operations model and user interface. Further information on configuring App Controller and using the user interface will be found in later chapters throughout this book.

App Controller resource configuring

After installing App Controller, a VMM administrator can log on using the App Controller web-based interface and connect a VMM server, clouds, Windows Azure subscription, third-party hosting, and network shares. Once the user has been authenticated, resources authorized for the user become accessible based on the user role assigned to the user.

Figure 1-13 shows an example of what a VMM administrator might see upon first logging on to the web-based interface after the App Controller installation process has finished. The Overview page includes Next Steps with a list of links for performing common tasks needed for configuring the App Controller environment. The navigation pane has a Settings workspace available for the VMM administrator to use. In the next chapter, we will walk through such steps as branding the App Controller website, connecting to VMM and Windows Azure, consuming services, and operating on deployment instances.

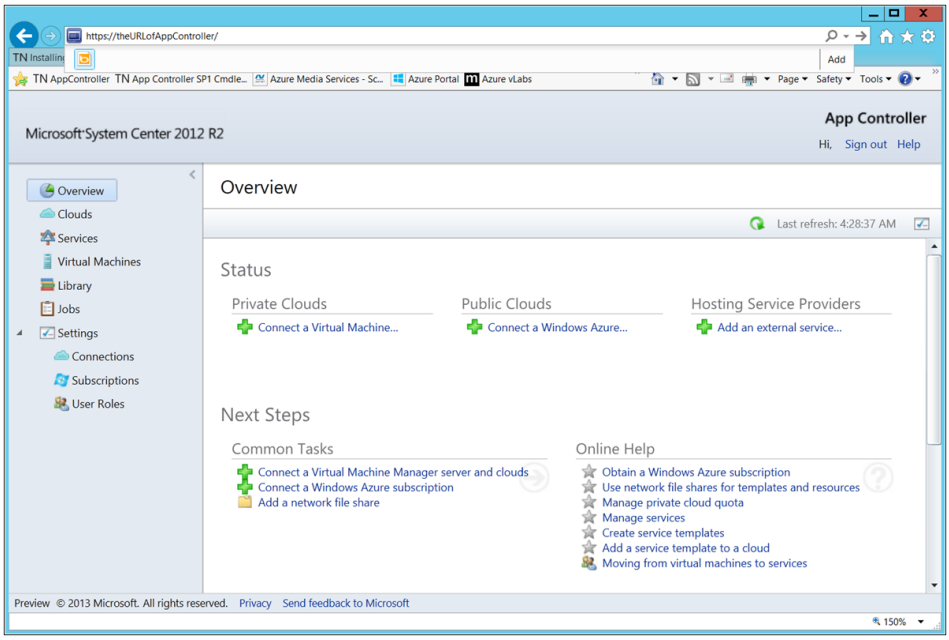


FIGURE 1-13 A view of the App Controller UI.

NOTE **Cloud service providers can provide multiple instances of App Controller targeting different users with different resources for different deployment scenarios to best serve the intended users.**

App Controller UI

As Figure 1-13 shows, the navigation pane for the App Controller web-based interface shares some similarity with the VMM admin console. But since App Controller is mainly a vehicle for consuming and managing resources, the web-based interface is used for deploying and operating on instances instead of for defining and configuring resources. From the top of the navigation pane, the workspaces are as follows:

- **Overview** This is a snapshot of the resources that are manageable based on what has been configured in the Settings workspace in the VMM administration console. Unlike in the VMM administration console, the Settings workspace is not visible to users in the App Controller web-based interface. In addition, the visibility and operability of resources like clouds, services, VMs, and library items are based on the user roles relevant to the authenticated user. The operations model for App Controller is to have only those resources authorized for the user to be visible so that the user can self-serve and deploy services with minimal IT support, if any.

- **Cloud** This is a logical container for the host services.
- **Services** This shows VMs that can be identified, managed, and operated as a single entity in order to deliver a particular line-of-business (LOB) application.
- **Virtual Machines** This shows deployed instances of VM templates. Here the individual VMs can be viewed and operated as individual objects.
- **Library** This is a repository for all of the resources available for creating virtual machines.
- **Jobs** This records a history of the jobs performed by App Controller.
- **Settings** This is where you can establish connections and access VMM and Windows Azure.

CHAPTER 2

Managing private clouds

Microsoft System Center 2012 R2 App Controller adds self-service management capabilities to your on-premises private clouds via an intuitive web-based GUI. In Chapter 1, we discussed that self-service management is a key ingredient for enabling a standardized approach for deploying applications in private and public clouds. In this chapter, we'll focus our attention specifically on private clouds. We'll walk through the steps of configuring and leveraging App Controller so that users, such as other administrators and developers on our IT teams, to whom we've delegated private cloud resources can easily deploy and manage workloads using nothing more than a web browser.

Specifically, we'll be targeting the following topics in this chapter:

- Which private clouds can be managed?
- App Controller and Virtual Machine Manager
- Preparing for self-service private cloud management
- Signing in at the portal
- Branding the portal experience
- Connecting to private clouds
- Adding a network file share
- Managing Run As accounts
- Deploying new workloads to a private cloud
- Managing Private Cloud Workloads
- Moving Files to/from Private Clouds

Which private clouds can be managed?

System Center 2012 R2 App Controller can provide self-service management for any private clouds that are defined using System Center 2012 R2 Virtual Machine Manager (VMM). Realistically, private clouds can include a heterogeneous mix of compute,

storage, and networking resources—after all, whose data center these days consists of only a single vendor solution? Luckily, VMM provides the ability to compose private clouds as pools of resources that leverage the multivendor environment that likely exists in your data center today. Such pools of resources could include:

- **Compute** Microsoft Hyper-V in Windows Server 2008 R2, Windows Server 2012, or Windows Server 2012 R2; VMware ESX/ESXi 4.1, 5.0 or 5.1 with VMware vCenter Server 4.1, 5.0 or 5.1; or Citrix XenServer 6.0.
- **Storage** Storage providers supporting the Storage Management Initiative Specification (SMI-S), a native Windows Management Instrumentation (WMI) Storage Management Provider (SMP) or Server Message Block (SMB) 3.0 storage management protocols.
- **Network** Top-of-rack (TOR) network switches that support the Common Information Model (CIM) standard for network device management; and virtual switch extensions, network switches, load balancers and Hyper-V Network Virtualization (HNV) gateways that offer a configuration provider module for System Center 2012 R2.

App Controller and Virtual Machine Manager

To use System Center 2012 R2 App Controller for self-service management of private clouds, you'll first need to install and configure System Center 2012 R2 VMM. After VMM is deployed and running, you'll then need use the VMM management console to build your private cloud fabric of compute, storage, and networking resources. You'll also need to define and delegate at least one private cloud as a pooled set of fabric resources. Optionally, you'll want to populate your VMM library with any profiles, VM templates and Application Service templates that you'll be leveraging via the App Controller web portal for deploying new workloads. Once you've completed these steps from the System Center 2012 R2 VMM management console, you're then ready to connect up App Controller to complete your private cloud management solution.

TIP **If you've not yet deployed System Center 2012 R2 VMM within your environment, take a break here to get it installed and configured first. For details on the process of building private clouds with VMM, be sure to leverage the information and step-by-step walkthroughs provided at *http://aka.ms/BuildYourCloud*. After you've installed System Center 2012 R2 VMM and have at least one private cloud defined and delegated, you can pick back up here to continue with the rest of this chapter.**

Preparing for self-service private cloud management

To enable self-service management of your private clouds with System Center 2012 R2 App Controller, you'll need to install App Controller in your data center environment. The server hardware and software requirements for System Center 2012 R2 App Controller are shown in Table 2-1. Note that App Controller deployments are supported on either physical hardware or as a virtual machine, as long as the system requirements are being met. Detailed server requirements are also available online at *http://aka.ms/SC2012AC-Requirements*.

TABLE 2-1 Server software and hardware requirements for System Center 2012 App Controller

Component	Minimum	Recommended
Processor	Pentium 4, 2 GHz (x64)	Dual-Processor, Dual-Core, 2.8 GHz (x64) or greater
RAM	1 GB	4 GB
Available hard disk space	512 MB	1 GB
Server operating system	Windows Server 2008 R2 Standard, Enterprise, or Datacenter edition	Windows Server 2012 / 2012 R2 Standard or Datacenter edition
.NET Framework	Microsoft .NET Framework 4.0 is required by System Center 2012 R2 App Controller. Microsoft .NET Framework 3.5.1 is required by the Windows PowerShell module for App Controller.	
Web Server (IIS)	The Web Server (IIS) role is required for hosting the App Controller management portal. It is recommend that you use the App Controller setup program to install and configure this role, rather than attempting to manually pre-install this role.	
SQL Server Database	SQL Server 2008 R2 Service Pack 2 Standard, Enterprise or Datacenter edition	SQL Server 2012 Service Pack 1 Standard or Enterprise edition running on a separate database server.
Active Directory	Any servers on which you are installing System Center 2012 R2 App Controller and/or Microsoft SQL Server must be members of an Active Directory domain.	
VMM Management Console	Any servers on which you are installing System Center 2012 R2 App Controller must have the management console software for System Center 2012 R2 VMM pre-installed. App Controller uses the VMM management console to programmatically communicate with VMM management servers for private cloud management. The complete steps for installing the VMM management console are available at *http://aka.ms/SC2012AC-VMMConsole*.	

TIP **Before launching the System Center 2012 R2 App Controller web console, ensure that your delegated private cloud management users have PCs that meet the following minimum requirements:**

- Operating system: Windows Vista or later version
- Web browser: A 32-bit browser that supports Microsoft Silverlight, such as Internet Explorer 8 or later

In terms of performance when managing private clouds, the recommended configuration in Table 2-1 has been tested to support the scalability limits presented in Table 2-2. As you can see from this table, using System Center 2012 R2 App Controller and VMM together, you can extend self-service private cloud management for even the largest virtualized data centers—up to 5 VMM management servers, 5,000 virtualization hosts, and 125,000 virtual machines!

TABLE 2-2 Private cloud performance and scale of System Center 2012 R2 App Controller

Task	Maximum Number
Managing System Center 2012 R2 VMM Management Servers	Maximum of 5 System Center 2012 R2 VMM Management Servers Each System Center 2012 R2 VMM Management Server can support up to 1,000 virtualization hosts and 25,000 virtual machines
Delegated users managing private cloud resources	Maximum of 75 concurrent users managing private cloud resources
Using jobs to deploy and manage private cloud resources	Maximum of 10,000 jobs can be run within a 24-hour period

For highly available App Controller installations, System Center 2012 R2 also supports the following high availability configurations:

- **Database Server** Install the database server as a clustered installation of SQL Server
- **App Controller** Install App Controller in a Highly Available Virtual Machine (HAVM) on a Hyper-V Host Cluster

With System Center 2012 R2, multiple App Controller servers can also be located behind a load balancer. Note that in a load-balanced configuration, each App Controller server will need to share a common encryption key. After installing the first App Controller server, you can export the encryption key by using the Export-SCACAesKey Windows PowerShell cmdlet. You will then provide this same exported encryption key when installing the other App Controller servers.

Once you've verified the server prerequisites, you're ready to install System Center 2012 R2 App Controller. Good news—the software installation process is very straightforward and can be accomplished in a short time. To assist in installing System Center 2012 R2 App Controller in your lab environment, take a break here and use these resources to build your lab server:

- **Download** System Center 2012 R2 Evaluation Kit (*http://aka.ms/SC2012AC-Download*)
- **Install** System Center 2012 R2 App Controller (*http://aka.ms/SC2012AC-Install*)

During the installation of System Center 2012 R2 App Controller, the setup program will automatically install .NET Framework 4.0 and the Web Server (IIS) role. In addition, on Windows Server 2008 R2 servers, .NET Framework 3.5.1 will also be automatically installed to support the Windows PowerShell module for App Controller. On Windows Server 2012 and later, .NET Framework 3.5.1 must be manually installed to use the Windows PowerShell module for App Controller.

Once you've completed the installation of System Center 2012 R2 App Controller, you can test the App Controller web portal by browsing to *https://<your_app_controller_server_name>*.

If you chose the option to generate a self-signed SSL certificate during the App Controller installation process for your lab, you might initially be presented with a certificate warning dialog box. Simply click the option to Continue To This Website to navigate to the App Controller login page. When installing System Center 2012 R2 App Controller in a production environment, it is recommended that you use a registered SSL certificate from a trusted certificate provider to eliminate this warning dialog box. In this case, the trusted certificate provider that you use could be either an internal trusted Certificate Authority (CA) within your organization or a publicly trusted CA. See *http://go.microsoft.com/fwlink/?LinkID=269988* for a current list of publicly trusted Root CAs that are distributed by Microsoft with Windows and Windows Server.

Signing in at the portal

When browsing to the App Controller portal page, you'll be prompted to sign in with Active Directory credentials to authorize your portal access, as shown in Figure 2-1.

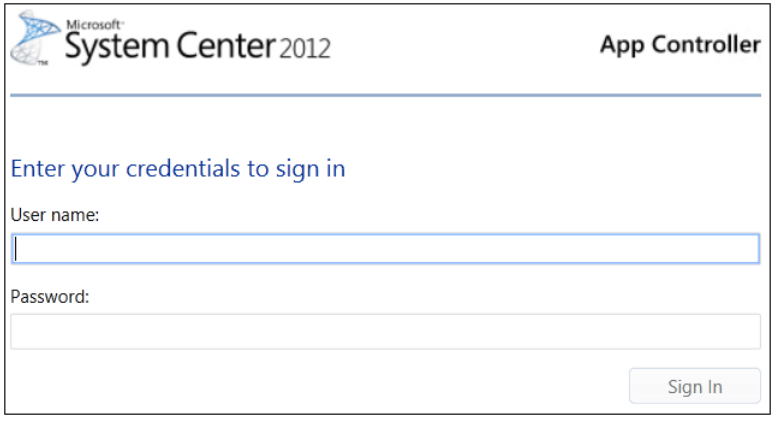


FIGURE 2-1 You can sign in to the System Center 2012 R2 App Controller portal.

At the App Controller sign-in page, enter the same Active Directory user credentials that you used when installing System Center 2012 R2 VMM and System Center 2012 R2 App Controller. Click the Sign In button to continue. Upon successful sign in, you will be presented with the System Center 2012 App Controller Overview portal page shown in Figure 2-2.

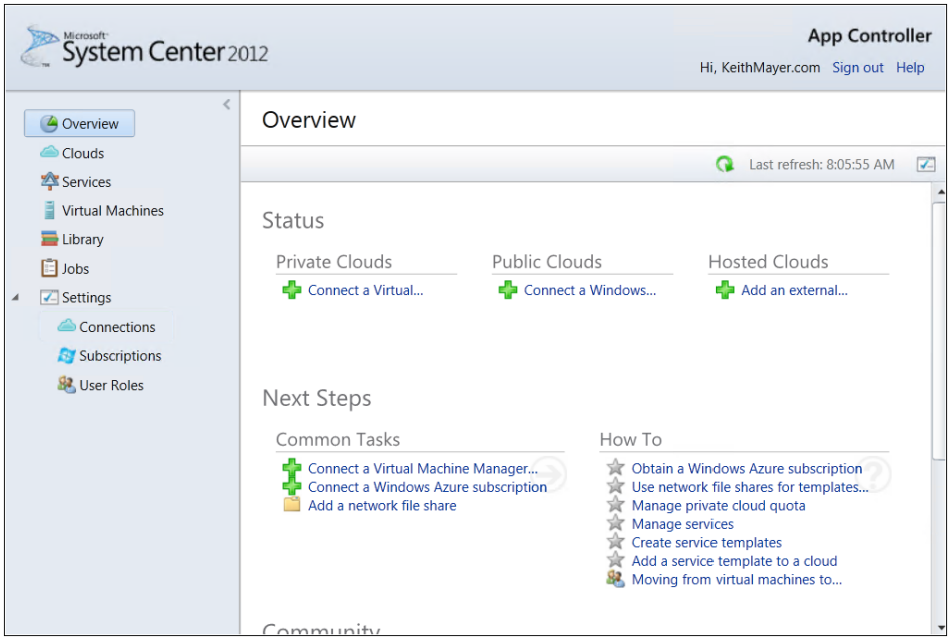


FIGURE 2-2 An example of the Overview page on the System Center 2012 App Controller portal.

Branding the portal experience

If desired, the System Center 2012 R2 App Controller portal pages can be easily branded for a particular organization. In Figure 2-2, the logos on the top navigation bar, called out in the figure with boxes, can be substituted for alternate graphics that align to internal branding for an IT organization by replacing specific files located in the C:\Program Files\Microsoft System Center 2012\App Controller\wwwroot folder on each App Controller server:

- **Top-left logo** Replace SC2012_WebHeaderLeft_AC.png with a 213px x 38px PNG file containing a transparent background
- **Top-right logo** Replace SC2012_WebHeaderRight_AC.png with a 108px x 16px PNG file containing a transparent background

Connecting to private clouds using App Controller

After signing in to the System Center 2012 R2 App Controller portal, you will need to connect App Controller to at least one System Center 2012 R2 VMM management server to begin managing private clouds via App Controller.

To add a VMM management server to the App Controller portal, complete the following steps:

1. Click the Connect A Virtual Machine Manager Server And Clouds link shown in Figure 2-3.

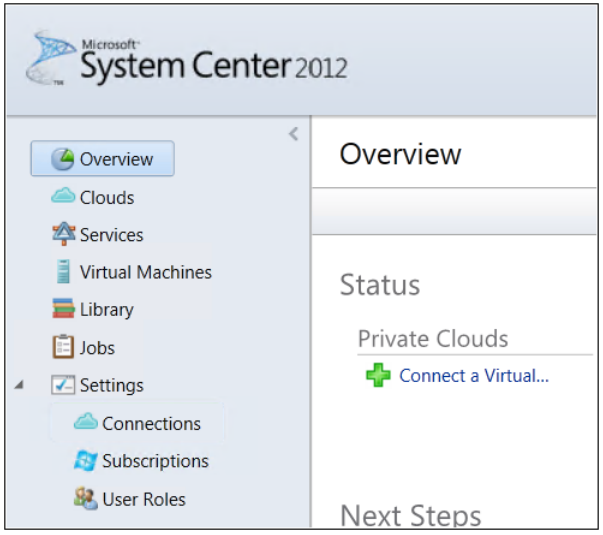


FIGURE 2-3 An example of adding a VMM management server to the App Controller portal.

2. In the Add A New VMM Connection dialog box that is displayed, provide the following information (see Figure 2-4):
   - **Connection Name** A display name for this VMM connection.
   - **Description** Text that describes the private clouds accessible via this VMM connection.
   - **Server Name** The Fully Qualified Domain Name (FQDN) of the System Center 2012 R2 VMM management server.
   - **Port** TCP port used for communication with the VMM management server (default = TCP/8100).
   - **Automatically Import SSL Certificates** Select this check box to import SSL certificates that App Controller will use for secure communication with the VMM server.

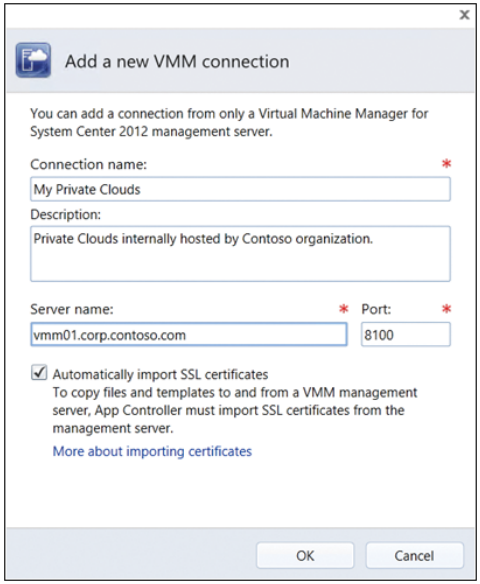

Add a new VMM connection
You can add a connection from only a Virtual Machine Manager for System Center 2012 management server.
Connection name:
My Private Clouds
Description:
Private Clouds internally hosted by Contoso organization.
Server name:
Port:
vmm01.corp.contoso.com
8100
Automatically import SSL certificates
To copy files and templates to and from a VMM management server, App Controller must import SSL certificates from the management server.
More about importing certificates
OK
Cancel

FIGURE 2-4 An example of adding a new VMM connection.

3. Click the OK button to add the new VMM connection.

   If you have been delegated more than one user role for managing private clouds in System Center 2012 R2 VMM, you will be prompted for the user role to use when managing private clouds via this connection as shown in Figure 2-5. If you are assigned only a single user role within VMM, this dialog box will not be displayed.

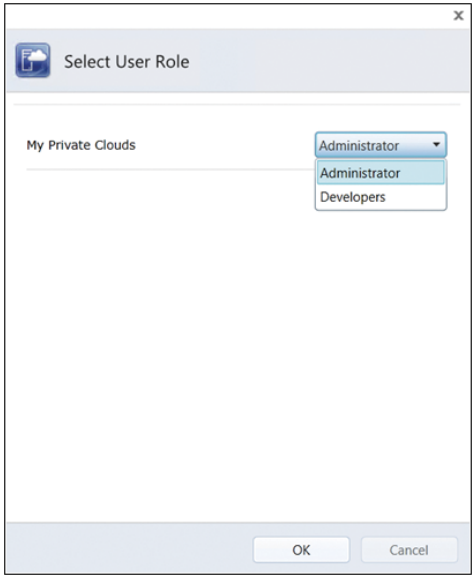


FIGURE 2-5 You can select user roles as shown here.

4. If prompted, select the desired user role and click the OK button.

The selected user role, and the associated delegated scope and allowed actions in VMM, will be used to determine the list of private clouds and actions that App Controller will make available from within the portal.

> ***NOTE*** **The selected user role is valid only for the current portal session. Each time you browse to the portal and open a new authenticated session, you will be prompted for the user role to use for that session if you are assigned to multiple roles for private clouds in VMM.**

After your connection is established to the VMM management server, a summarized status of the private clouds and virtual machines currently manageable via this connection will be displayed on the Overview page as shown in Figure 2-6.

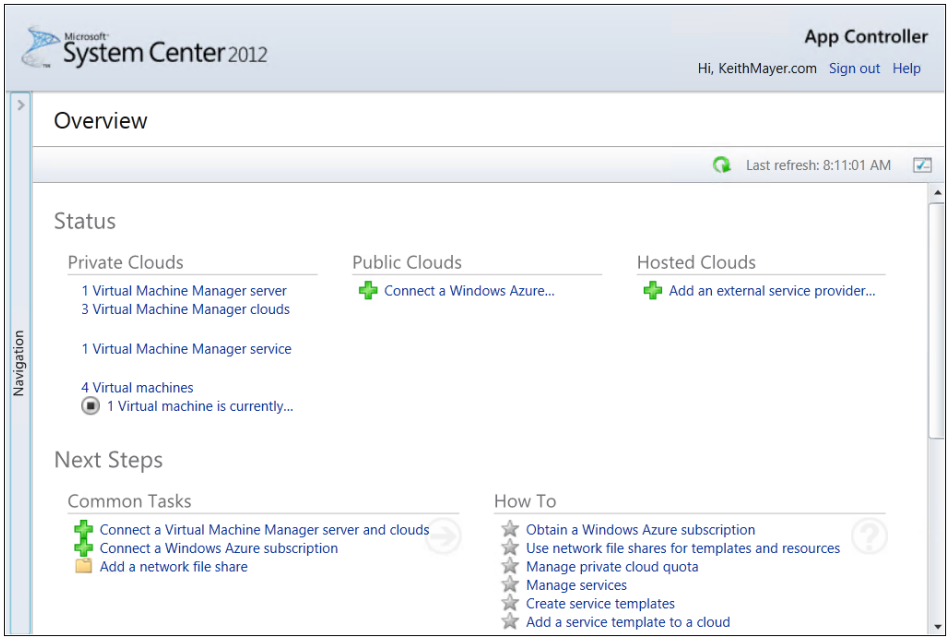


FIGURE 2-6 The Overview page showing the private clouds.

At this point, you've successfully connected to private clouds using System Center 2012 R2 App Controller.

After a VMM connection is established, to manage, add, or remove VMM connections from the App Controller portal, you may use the Settings\Connections portal page, as shown in Figure 2-7.

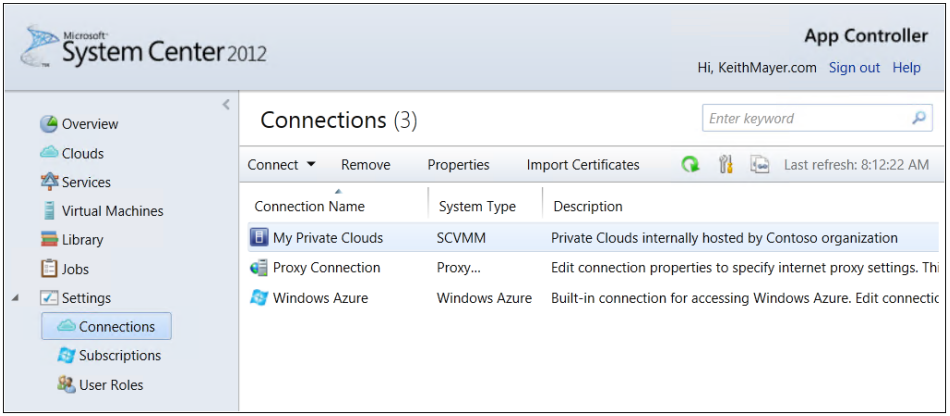


FIGURE 2-7 You can use the Connections page when managing private cloud connections.

Adding a network file share to App Controller

In addition to connecting to one or more VMM management servers, App Controller also provides the ability to connect to one or more network file shares to which you have previously been granted permissions. Network file shares are useful in App Controller when copying virtual machine files from other locations to/from a VMM library for deployment within a private cloud.

> ***NOTE*** **If files will be copied to/from an added file share via the App Controller portal, the machine account for each App Controller server must also be granted Full Control permissions to each added file share.**

To add a new network file share to the App Controller portal, complete the following steps:

1. Click the Add A Network File Share link under Common Tasks in the Next Steps section of the Overview portal page as shown in Figure 2-8.

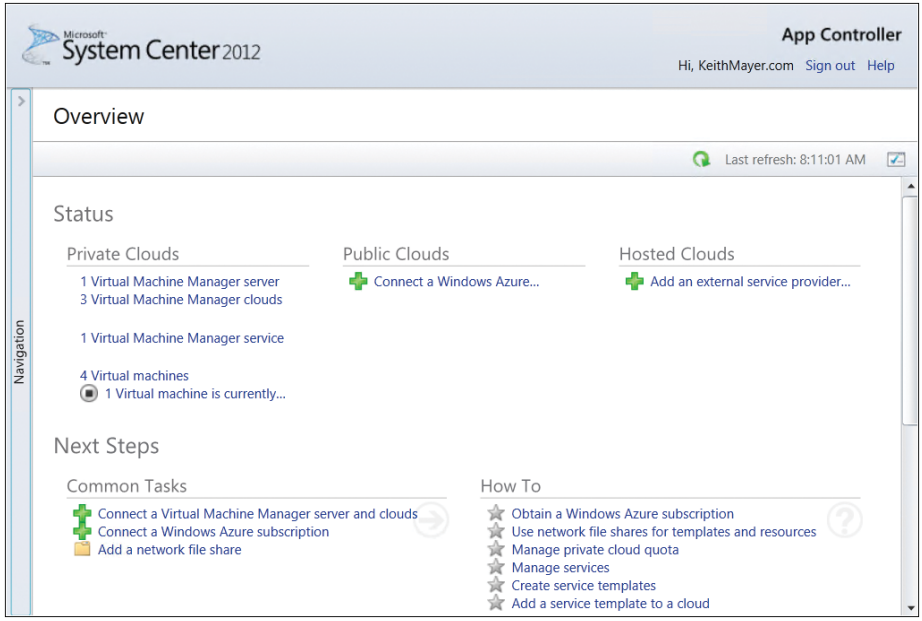


FIGURE 2-8 You can add a network file share from the Overview page.

2. In the Add A Network Shared Folder dialog box, enter the share path in UNC (for example, \\server\share) format and click OK as shown in Figure 2-9.

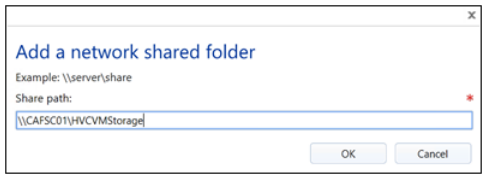


FIGURE 2-9 You can see the path in the Add A Network Shared Folder dialog box.

3. After a network file share has been added to the App Controller portal, it can be accessed by navigating to the Library page within the portal, as shown in Figure 2-10.

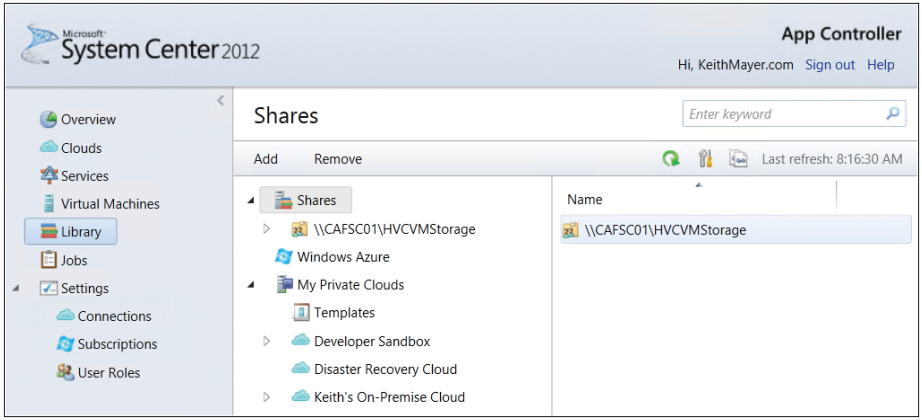


FIGURE 2-10 You can view the Shares within the Library.

On the Library portal page, network file shares and VMM libraries that were configured during VMM installation are displayed. From this portal page, you can browse the contents of each network file share and VMM library, copy files to/from each location, and add or remove network file shares. Later in this chapter, we'll step through the process of copying files to/from network file shares to prepare for deploying new workloads within a private cloud.

Managing Run As accounts

When deploying and managing application workloads, administrators regularly encounter several sets of administrative credentials and service account credentials that are needed to properly configure applications for connecting with underlying operating system resources, databases, and other application components. In System Center 2012 R2, the handling of administrative credentials is both simplified and standardized through the use of Run As accounts. Rather than being forced to remember a long list of administrative usernames and passwords for each application, administrators can instead create one set of Run As accounts that contain the necessary credentials. During deployment time, the appropriate Run As accounts can be selected, and System Center 2012 R2 will automatically supply the saved usernames and passwords that are associated with the selected accounts. Run As accounts also provide an effective means of delegating access to other IT administrators or developers for leveraging these credentials when deploying their private cloud workloads without needing to reveal the specific username and password values to these self-service IT users.

To manage the current list of Run As accounts for a set of private clouds, complete the following steps:

1. Click the Clouds page in the App Controller portal and right-click one of the displayed private clouds and click the Manage Run As Accounts option shown in Figure 2-11.

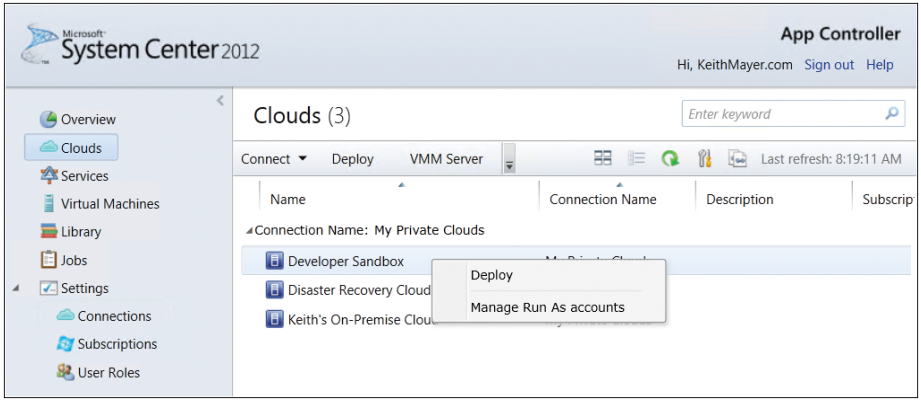


FIGURE 2-11 An example of managing a Run As account.

2. From the right-click menu, click Manage Run As Accounts to display the Create, Edit Or Delete Run As Accounts page as shown in Figure 2-12.

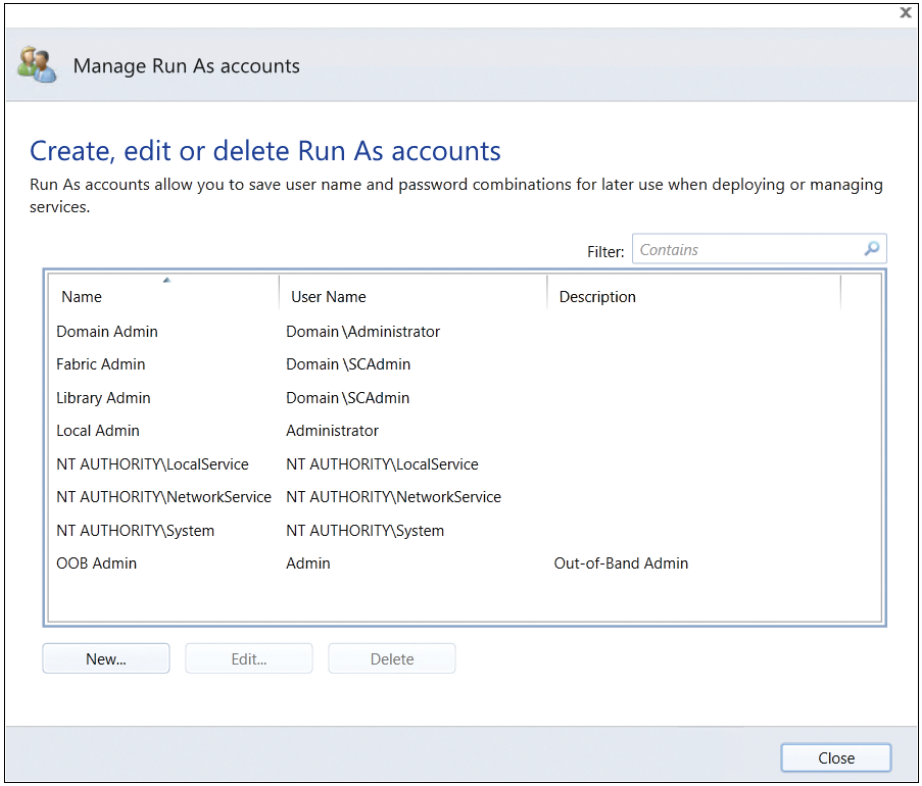


FIGURE 2-12 You can create, edit, or delete Run As accounts.

3. To define a new Run As account, click Create to navigate to the New Run As Account page, as displayed in Figure 2-13.

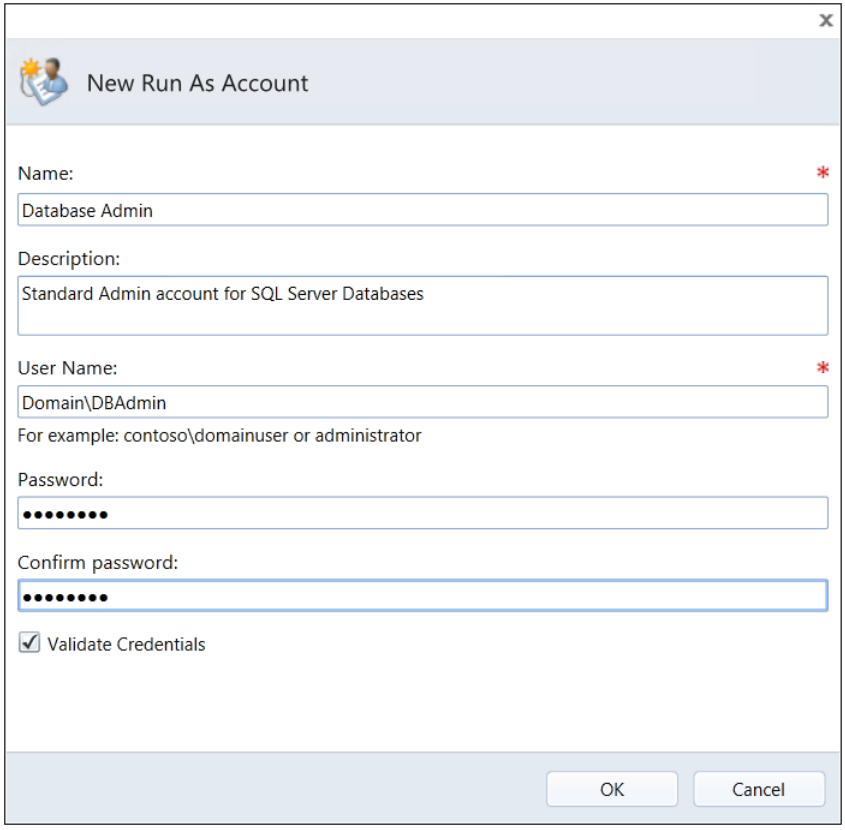


FIGURE 2-13 You can specify details in the New Run As Account dialog box.

4. Enter the username and password of an existing administrative account or service account, and then click OK to save these credentials as a new Run As account. Click Close when finished managing Run As accounts.

Deploying new workloads to private clouds

It is easy to deploy new application workloads to private clouds from the System Center 2012 R2 App Controller portal. During deployment, existing private clouds, delegated access rights, and template resources that were previously defined within System Center 2012 R2 VMM can be leveraged to safely extend deployment operations to authorized self-service IT users. To learn more about defining these configuration items in System Center 2012 R2 VMM, see the following modules in the Build Your Private Cloud (*http://aka.ms/BuildYourCloud*) online series:

- **Module 8** Creating and Delegating Private Clouds with System Center 2012 VMM

- **Module 9** Deploying and Managing Private Cloud Applications with System Center 2012 VMM

To deploy a new application workload to an existing private cloud using System Center 2012 R2 App Controller, complete the following steps:

1. Click the Clouds page and then right-click the private cloud to which the new application workload should be deployed and select the Deploy option shown in Figure 2-14.

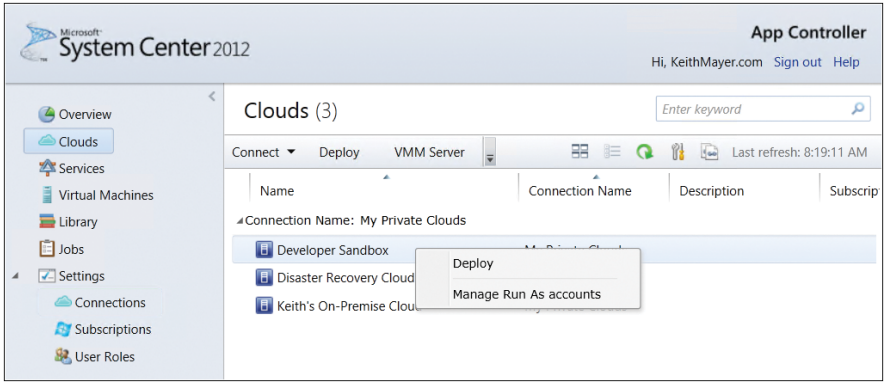


FIGURE 2-14 A new workload can be deployed to a private cloud.

2. From the right-click menu, select Deploy to launch the New Deployment dialog box, as shown in Figure 2-15.

FIGURE 2-15 Use the New Deployment dialog box to select a template.

3. On the New Deployment dialog box, click Select A Template and select the appropriate VM Template or Service Template previously defined in System Center 2012 R2 VMM (see Figure 2-16). VM Templates are used to specify a template configuration for a single VM being deployed to a private cloud, whereas Service Templates can include a template configuration for more complex multi-tier applications that can involve multiple virtual machines, applications, virtual networks, and load balancers as part of a single template.

TIP **You'll also find a Copy URL To Clipboard link located at the top-right corner of the New Deployment page. This link is useful for copying the direct link to this page and sharing it with other authorized users as a shortcut for deploying additional workloads to this same private cloud.**

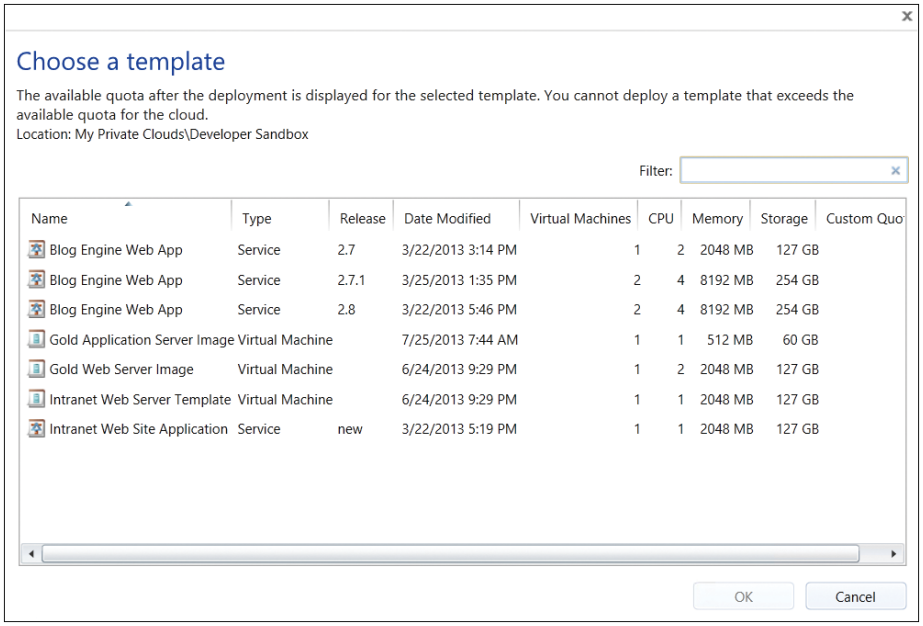

Choose a template
The available quota after the deployment is displayed for the selected template. You cannot deploy a template that exceeds the available quota for the cloud.
Location: My Private Clouds\Developer Sandbox
Filter:
Name
Type
Release
Date Modified
Virtual Machines
CPU
Memory
Storage
Custom Quo
Blog Engine Web App Service 2.7 3/22/2013 3:14 PM 1 2 2048 MB 127 GB
Blog Engine Web App Service 2.7.1 3/25/2013 1:35 PM 2 4 8192 MB 254 GB
Blog Engine Web App Service 2.8 3/22/2013 5:46 PM 2 4 8192 MB 254 GB
Gold Application Server Image Virtual Machine 7/25/2013 7:44 AM 1 1 512 MB 60 GB
Gold Web Server Image Virtual Machine 6/24/2013 9:29 PM 1 2 2048 MB 127 GB
Intranet Web Server Template Virtual Machine 6/24/2013 9:29 PM 1 1 2048 MB 127 GB
Intranet Web Site Application Service new 3/22/2013 5:19 PM 1 1 2048 MB 127 GB
OK
Cancel

FIGURE 2-16 You can select a template from the Choose A Template dialog box.

4. After selecting the desired template for deploying a new workload, click OK. This will return to the prior New Deployment dialog box where you'll be presented with options to configure the settings for this new deployment, as shown in Figure 2-17.

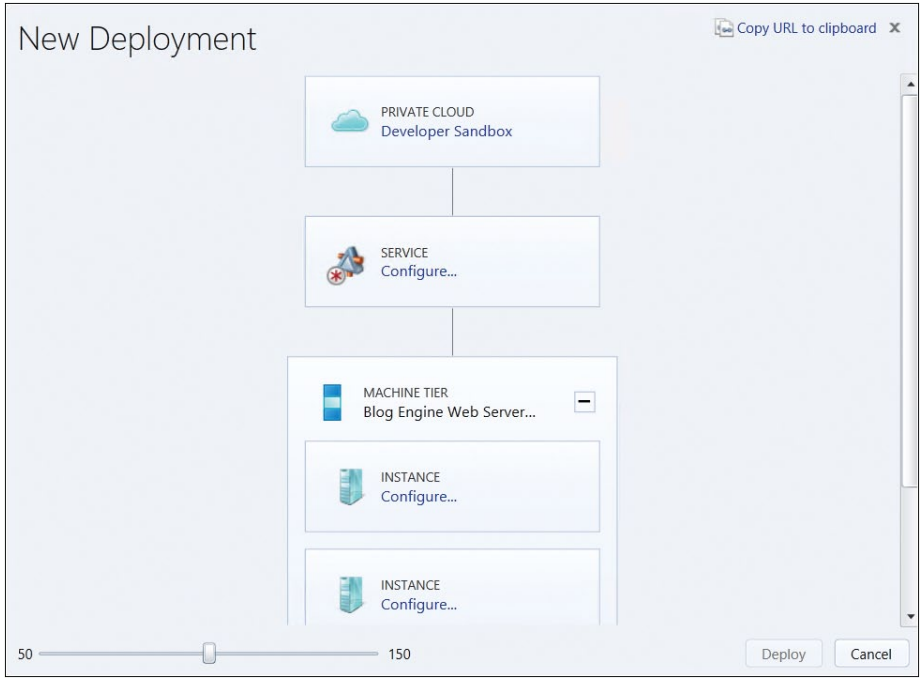


FIGURE 2-17 Use the Configure Settings from the New Deployment dialog box to configure the deployment.

5. On the New Deployment dialog box, click Configure in the SERVICE box to configure the general configuration properties for this new application workload. This will display the Properties page for the new service, or application workload, being deployed to the selected private cloud (see Figure 2-18).

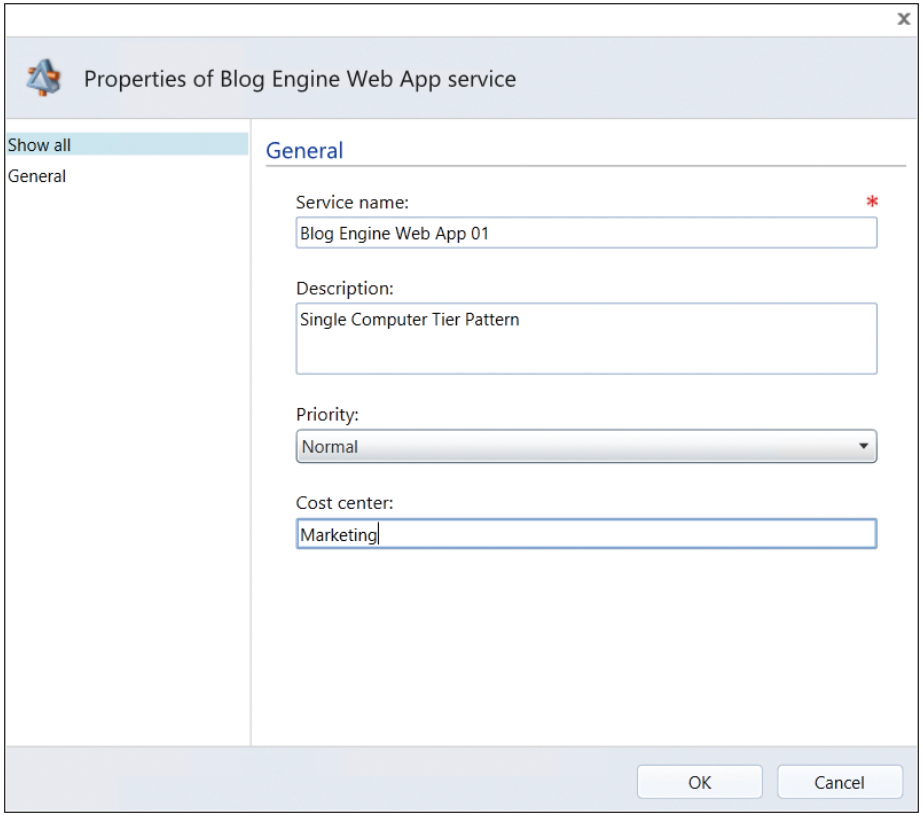


FIGURE 2-18 Configure the properties of the new service.

6. In the Properties page, enter a Service name for the new service being deployed, and optionally assign a Description, Priority, and Cost Center. Click OK to save this configuration information for the Service and return to the New Deployment dialog box, as displayed in Figure 2-19.

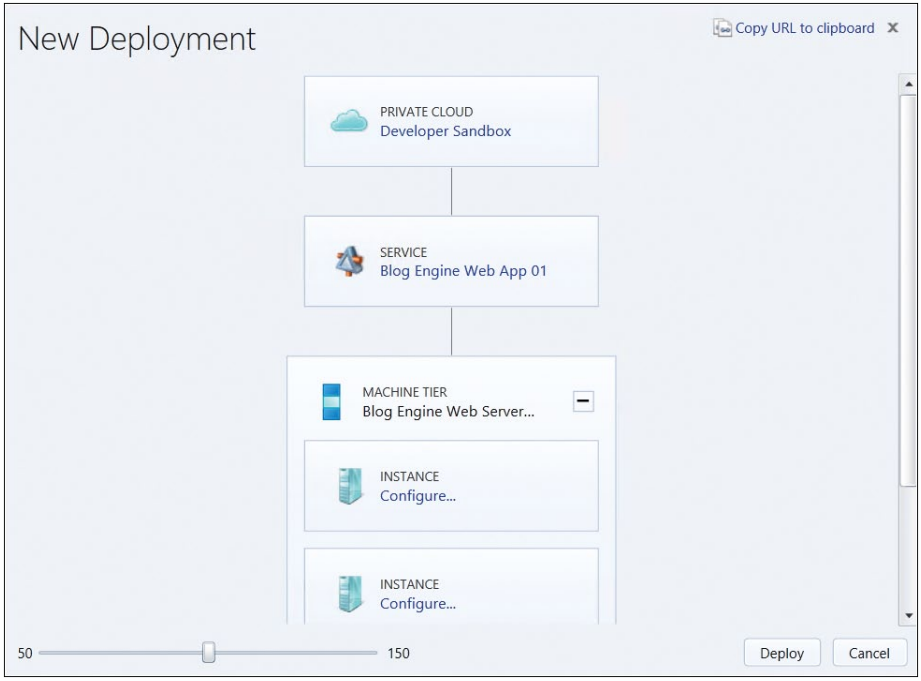


FIGURE 2-19 A summary of the new deployment showing the configured service.

7. Similarly, to configure each VM in each Machine Tier of the service being deployed, click each Configure in the INSTANCE box in the New Deployment dialog box to enter virtual machine configuration settings, if required by the template being used for deployment (see Figure 2-20).

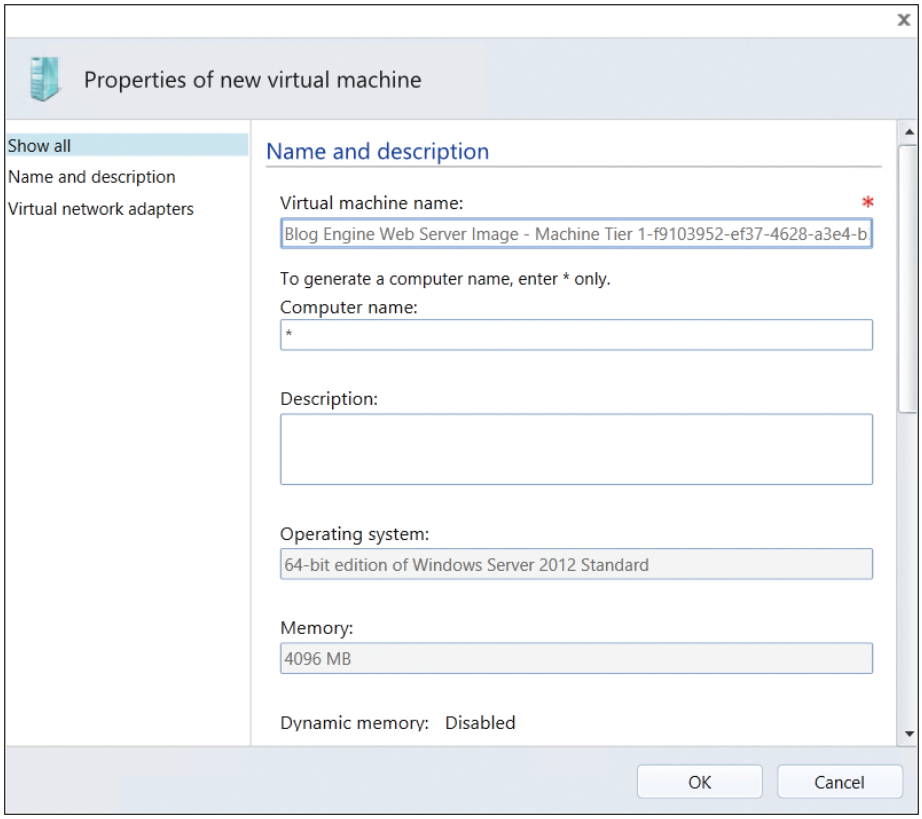


FIGURE 2-20 You can set the name and description in the Properties Of New Virtual Machine page.

8. For the Service Template that is being used in this example, all VM properties have been completed automatically by the template, so there's no additional configuration information needed. Click OK to return to the prior New Deployment page.
9. To deploy the new application workload to the selected private cloud, click Deploy. Depending on the complexity of the Service Template or VM Template being deployed to a private cloud, the deployment process can require several minutes to complete.

While the deployment is being processed, the Jobs page on the App Controller portal can be used to confirm the status of in-progress jobs. When all jobs associated with the new deployment are displayed with a Completed status, the new application workload will have been successfully deployed, as shown in Figure 2-21.

After the successful deployment of the application workload to the selected private cloud, the Services and Virtual Machines pages in the App Controller portal can be used to confirm workload status and manage the deployed workloads, as shown in Figure 2-22 and Figure 2-23.

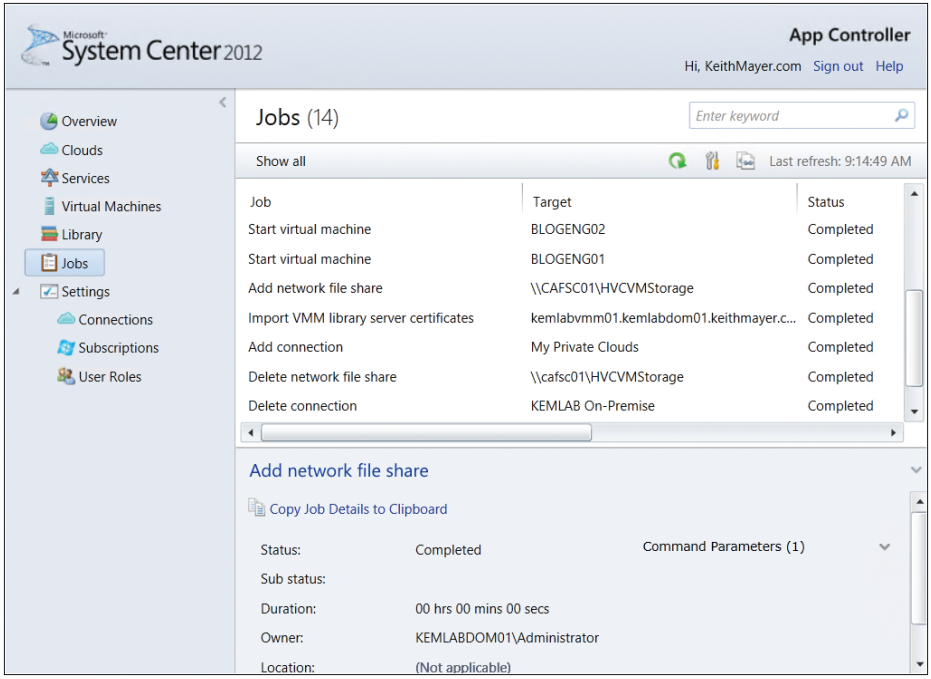


FIGURE 2-21 The App Controller Portal allows you to view the Jobs page.

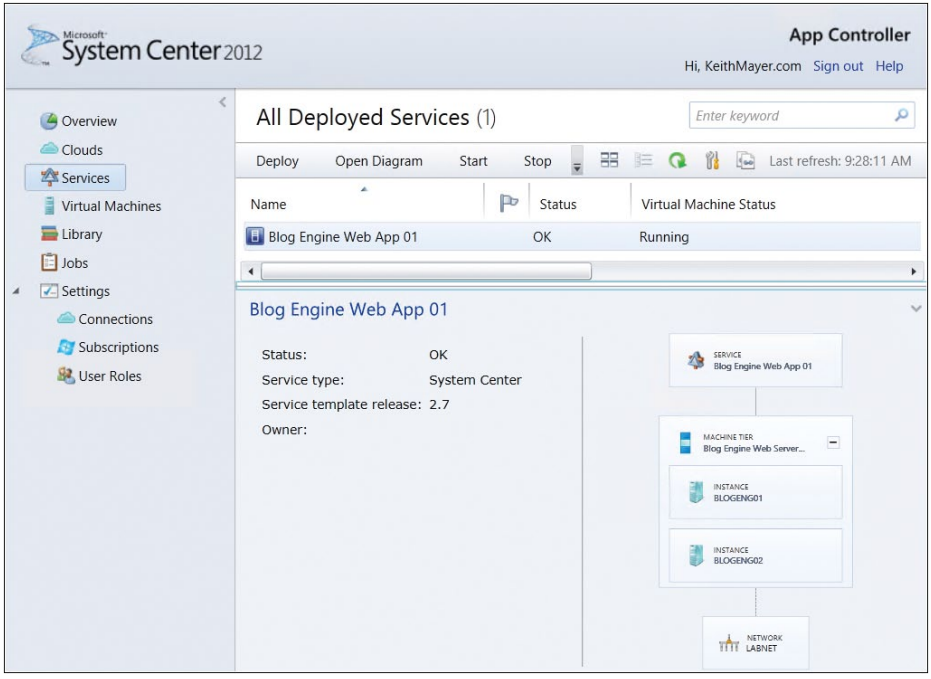


FIGURE 2-22 You can view all Deployed Services in a private cloud.

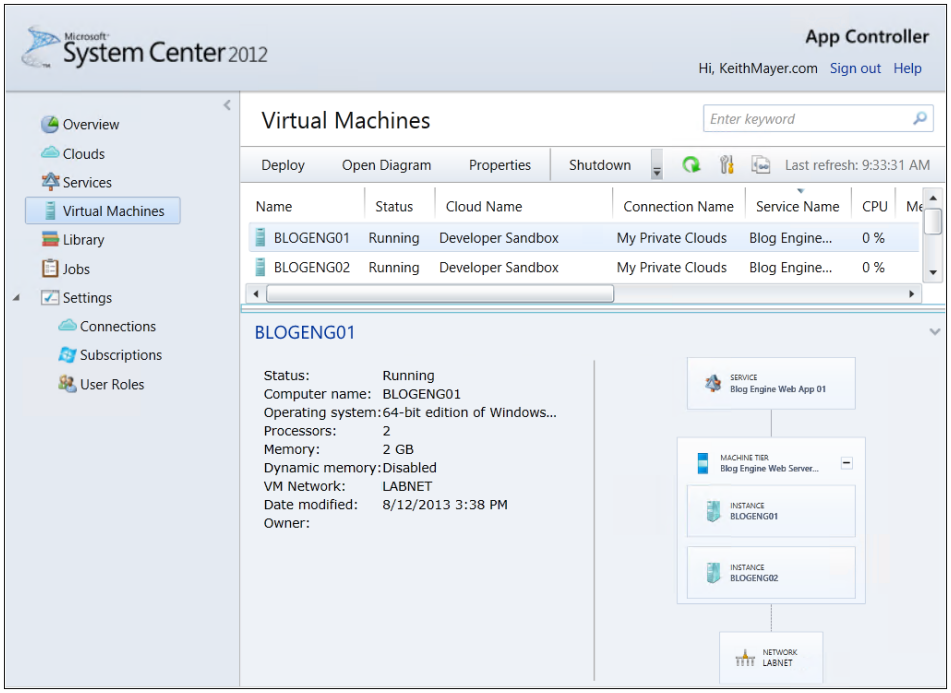


FIGURE 2-23 You can view the deployed Virtual Machine instances in a private cloud.

Managing private cloud workloads

After successful deployment of new workloads to a private cloud, these workloads can be managed from the Services and Virtual Machines pages in the App Controller portal. By right-clicking existing deployed Services or VMs on these pages, a set of convenient management actions is presented (see Figure 2-24).

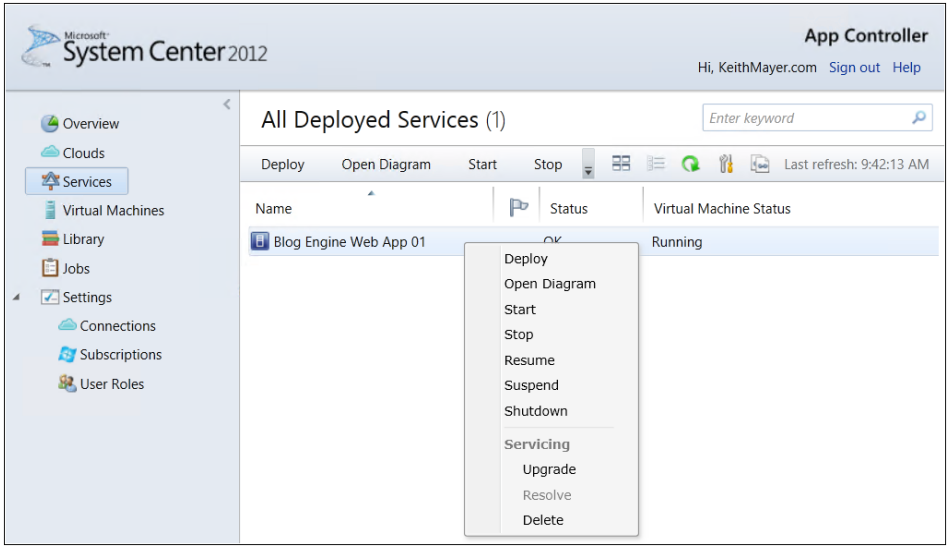


FIGURE 2-24 You can manage services in a private cloud by right-clicking the deployed service.

Right-clicking an existing deployed service on the Services page in the App Controller portal presents a list of actions for managing all components of a service as a single unit. Actions are available to Start, Stop, Suspend, Resume, and Shutdown all VMs associated with this service in a coordinated manner (see Figure 2-25).

In addition, when upgrades to this service need to be deployed, an Upgrade action is presented that permits a new version of a Service Template to be selected for orchestrated deployment across all VMs in that service. This is a powerful feature that permits workload upgrades to be rolled forward and/or rolled backward as needed to support the lifecycle of a deployed application through pilot testing, production roll-outs, patches, and ongoing code revisions.

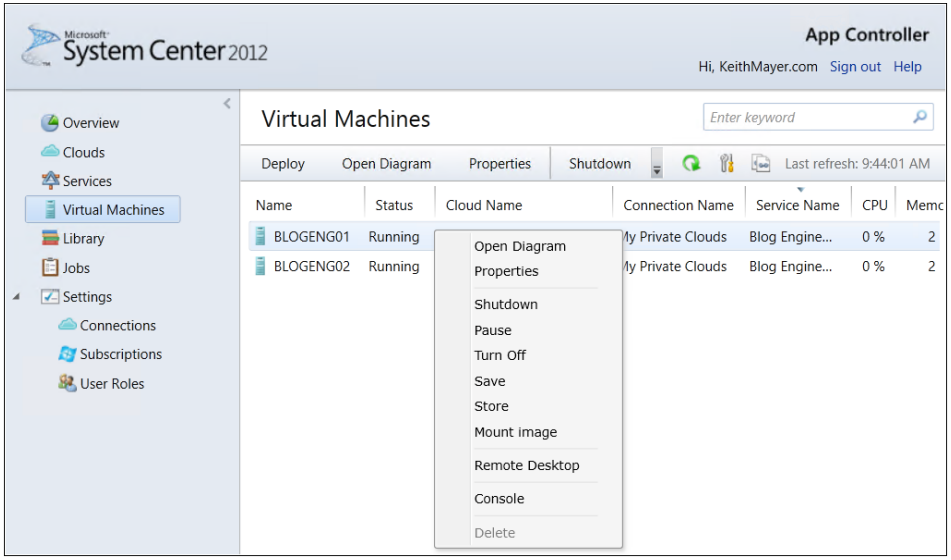


FIGURE 2-25 You can right-click a VM name to manage VMs in a private cloud.

Right-clicking an existing VM on the Virtual Machines page in the App Controller portal presents a list of actions for managing individual VMs, which could be a subset of a larger service that has been deployed or could simply be discrete VMs that were directly deployed to a private cloud using a VM Template. Actions are available to Start/Shutdown, Pause/Resume, Turn On/Off, and Save VMs.

In addition, two remote control options are available for interacting with the operating system and applications running within each VM:

- **Console** Used to establish a virtual KVM (Keyboard, Video, and Mouse) connection to the virtual machine's console from within a webpage
- **Remote Desktop** Used to establish a remote console connection via the Remote Desktop Protocol (RDP) using Remote Desktop Connection (RDC) client software, assuming the VM is in a state that can accept incoming RDP sessions

Other useful items on this menu include:

- **Mount image** Used to make an ISO image available to a VM for virtual CD/DVD support
- **Store** Used to store a VM in a VMM library for use as a VM template to support future workload deployments
- **Properties** Used to view and/or modify the properties, if permitted, of a particular VM

Moving files to/from private clouds

When managing private clouds from the App Controller portal, certain tasks might require moving files between network file shares and private cloud VMM libraries. Developer and test lab environments present a common scenario where files might need to be moved in this manner—developer and/or test lab VMs can exist outside of the context of private clouds managed by System Center 2012 R2, and to deploy these VMs to one or more private clouds, these files first need to be moved to a VMM library.

It is easy to move files between network file shares and VMM libraries via the Library page in the App Controller portal, as shown in Figure 2-26.

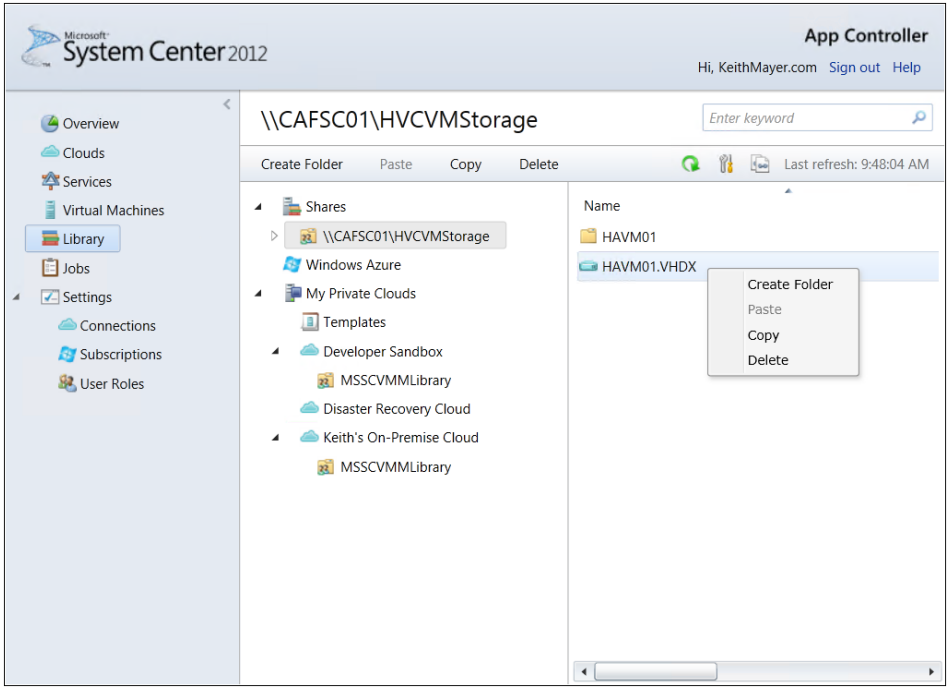


FIGURE 2-26 You can move files to or from private clouds using VMM libraries.

To move files, such as VM virtual hard disks (for example, .VHD or .VHDX files), from a network file share to a VMM library, select the file from the network file share that was previously added to the App Controller portal and right-click. On the right-click menu, select Copy. After the files have been selected for a copy operation, click the appropriate VMM library folder and then click Paste on the top toolbar.

To move files from a VMM library to a network file share, simply reverse this process by selecting the files for a copy operation from within the appropriate VMM library folder, and then paste into the desired network file share.

CHAPTER 3

Managing public clouds

In Chapter 2, "Managing private clouds," we discussed the capabilities of System Center 2012 R2 App Controller for supporting on-premises private cloud management. In addition to managing private clouds, App Controller also extends self-service management capabilities to public clouds, such as Windows Azure. Being able to provision and manage resources, whether serviced by a private or public cloud, from a single intuitive, web-based portal unifies cloud management with a consistent set of processes for managing application workloads regardless of the location of those workloads.

In this chapter, we'll look at the process for enterprise management of Windows Azure public cloud subscriptions with App Controller as we progress through the following topics:

- Why public cloud?
- Introducing Windows Azure
- Managing Windows Azure with the Windows Azure Management Portal
- Managing Windows Azure with System Center 2012 R2 App Controller
- Preparing for self-service public cloud management
- Creating a self-signed management certificate
- Uploading a management certificate to Windows Azure
- Connecting to public clouds
- Delegating access to public clouds
- Creating a Windows Azure storage account
- Deploying new workloads to a public cloud
- Managing public cloud workloads
- Managing files, disks, and images in public clouds

Why public cloud?

Public cloud platforms, such as Windows Azure, provide globally connected cloud-based data centers that have elastic capacities for delivering extreme volumes of compute, networking, and storage resources on an as-needed basis. This extreme elasticity of resources is attractive to many organizations for augmenting their on-premises data center with additional on-demand capacity. Otherwise, these organizations are finding that they are increasingly limited by the finite capacity and location of their own data centers alone.

Some of the key scenarios where organizations have seen great successes in leveraging public cloud platforms today include applications that have the following characteristics:

- "On and Off" workloads, such as on-demand development and test lab environments
- Fast capacity growth, such as evaluating and piloting new applications that might need to move quickly to production scale-out
- Unpredictable needs, such as off-site disaster recovery solutions for on-premises applications
- "Bursty" application requests, such as Internet-facing web applications and web services
- Applications requiring high scale of resources for short periods of time, such as batch processing applications and high-performance computing (HPC) calculations.

Of course, there are many different scenarios that make sense for leveraging public cloud platforms, but the scenarios we've listed are some of the most common application "patterns" to help you get started in thinking about public clouds in terms of your applications.

Introducing Windows Azure

Windows Azure is Microsoft's public cloud platform that provides on-demand, self-service provisioning of cloud resources across a global network of Microsoft cloud data centers. Windows Azure is the only cloud platform today that supports the flexibility of all three of the common cloud computing models in the industry (see Figure 3-1):

- **Infrastructure as a Service (IaaS)** Provision, manage, and migrate cloud-based infrastructure resources, such as virtual machines (VMs), virtual networks, and cloud-based storage. IaaS is ideal for migrating existing on-premises applications to a cloud platform without needing to rewrite application code.
- **Platform as a Service (PaaS)** Develop and deploy new custom applications designed to take advantage of the scale of cloud platforms. PaaS is ideal for quickly developing new applications or enhancements to existing applications on a cloud platform. PaaS generally provides better cloud cost economics than IaaS for newly developed applications due to an optimized, managed virtual environment that provides the application programming interfaces (APIs) and underlying operating systems.

- **Software as a Service (SaaS)** Quickly provision new packaged software in the cloud. SaaS is ideal for making prepackaged software, such as third-party web applications, available to users on a cloud platform. For these prepackaged applications, SaaS generally provides the lowest cloud costs due to optimized and managed software applications.

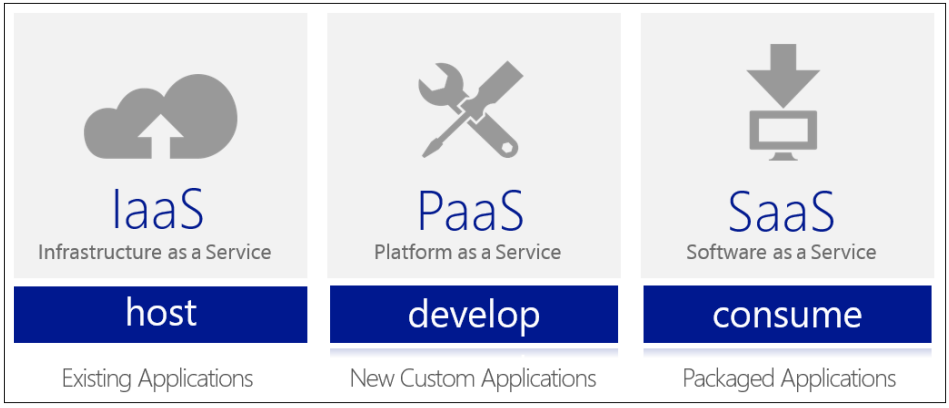


FIGURE 3-1 Three common cloud computing models.

By supporting all three of the common cloud computing models, Windows Azure provides a public cloud platform that can be leveraged throughout an application's lifecycle for initially migrating applications to the IaaS model, enhancing applications with the PaaS model, and ultimately consuming those applications as mature, packaged cloud applications with the SaaS model.

You can learn more about the capabilities of Windows Azure and sign up for a free trial subscription for hands-on evaluation at *http://aka.ms/WindowsAzureFreeTrial*.

Managing Windows Azure with the Windows Azure Management Portal

As a self-service and on-demand cloud platform, Windows Azure subscriptions can be conveniently managed on an individual basis via the Windows Azure Management Portal. The Windows Azure Management Portal is a web-based portal accessible from any modern web browser supporting HTML5 and JavaScript. Via this portal, the complete portfolio of Windows Azure cloud platform services across IaaS, PaaS, and SaaS models can be directly managed by a subscribing user. The Windows Azure Management Portal can be accessed at *http://manage.windowsazure.com*.

After signing in to the Windows Azure Management Portal with a Microsoft Account (formerly known as a Microsoft Windows Live ID) that is associated with a valid Windows Azure paid subscription or trial subscription, the portal page shown in Figure 3-2 will be

displayed. From this portal page, the blue navigation bar located on the left edge of the page can be used to select a particular Windows Azure cloud service to be provisioned or managed, such as Virtual Machines, Networks, or Storage. Once a particular service is selected, the page for that particular service will be displayed and the black toolbar located at the bottom edge of the page can be used to provision new cloud resources using the +NEW toolbar button, or manage existing cloud resources using the other toolbar buttons.

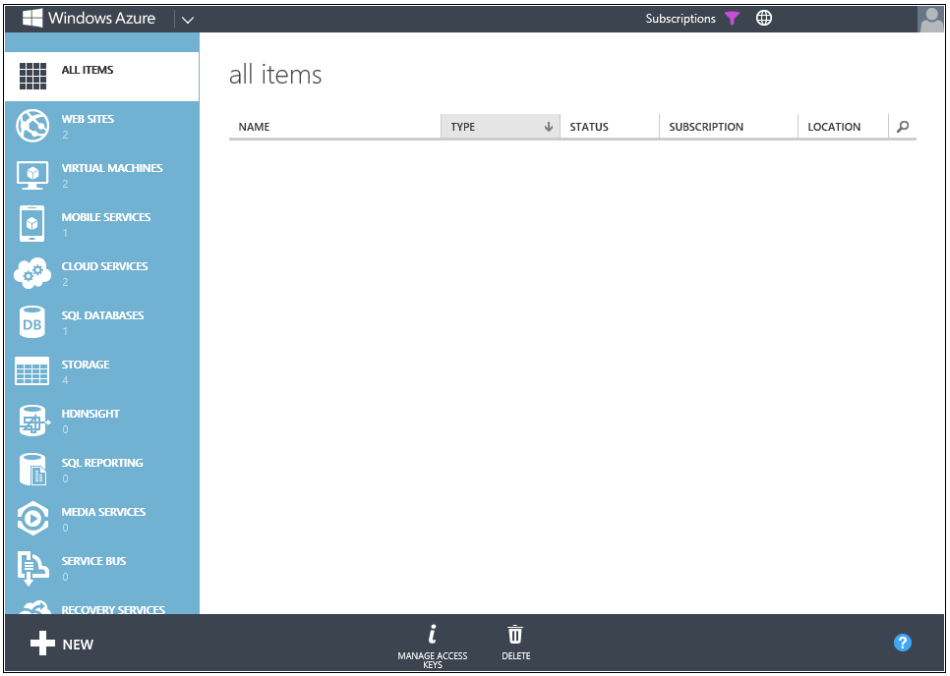


FIGURE 3-2 The Windows Azure Management Portal.

To learn more about Windows Azure infrastructure services, be sure to leverage the free online content available as part of the Early Experts cloud program. You'll find this content available at *http://aka.ms/EarlyExpertsCloud*.

Managing Windows Azure with System Center 2012 R2 App Controller

As we discussed in the prior section, managing Windows Azure via the web-based Windows Azure Management Portal is quite easy and straightforward for individual users; however, access to this management portal can be challenging to manage across an entire

organization. The Windows Azure Management Portal leverages Microsoft Accounts (formerly Microsoft Windows Live IDs) for authentication, and in larger organizations this can make it challenging to provide several IT administrators or developers with delegated access to Windows Azure. Additionally, most organizations require the ability to assign only the self-service access permissions needed to provision and manage cloud services to authorized members of an IT, development, or business unit team, rather than granting full administrative access to all members of the organization.

Preparing for self-service public cloud management

To manage public clouds via App Controller, you'll need at least one Windows Azure subscription. If you don't currently have an active subscription for Windows Azure, you can sign up for a free trial subscription, which is suitable for evaluating Windows Azure in a lab environment with App Controller. You can sign-up for your free Windows Azure trial subscription at *http://aka.ms/WindowsAzureFreeTrial*.

After you have access to an active Windows Azure subscription, you'll next need to set up a management certificate. To maintain a secure connection between App Controller and Windows Azure, a management certificate is used to authenticate the connection. In a production environment, you'll typically want to provision this management certificate via an internal Public Key Infrastructure (PKI) so that certificates can be easily managed and renewed on a centralized basis. For production environments, Windows Server 2012 can be configured with the Active Directory Certificate Services (AD CS) role to host an internal PKI, and while this topic is outside the scope of this book, you can find additional details on planning and deployment of AD CS in the Microsoft TechNet library at *http://aka.ms/SC2012AC-ADCS*. Alternatively, in a lab environment, you can also quickly create a self-signed management certificate using the process included in the next section of this chapter.

Creating a self-signed management certificate

In a lab environment, you can quickly set up a self-signed certificate from the App Controller server console using the IIS Manager tool. This self-signed certificate does not provide the same level of centralized trust and management as a full PKI, but it can be used to provide a secure set of credentials suitable for evaluating public clouds with App Controller. At the App Controller server console, you can launch the IIS Manager tool by selecting Tools and then Internet Information Services (IIS) Manager from the Windows Server 2012 Server Manager tool, as shown in Figure 3-3.

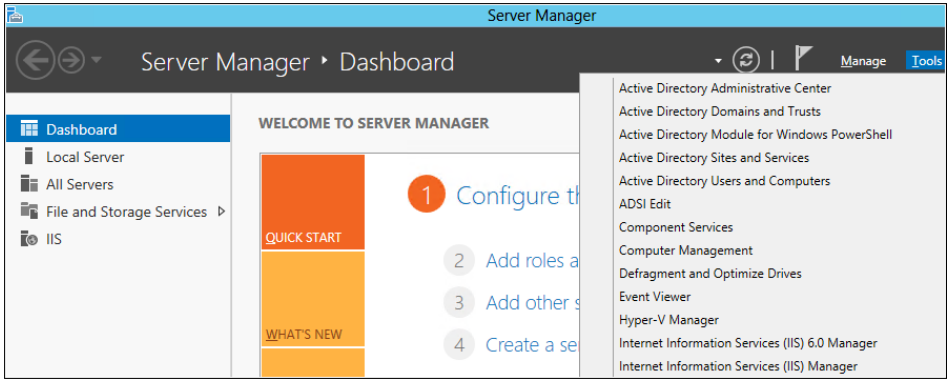


FIGURE 3-3 You can launch IIS Manager from Server Manager.

After the IIS Manager tool has launched, double-click Server Certificates at the bottom of the IIS Manager Home page to create and manage certificates as shown in Figure 3-4.

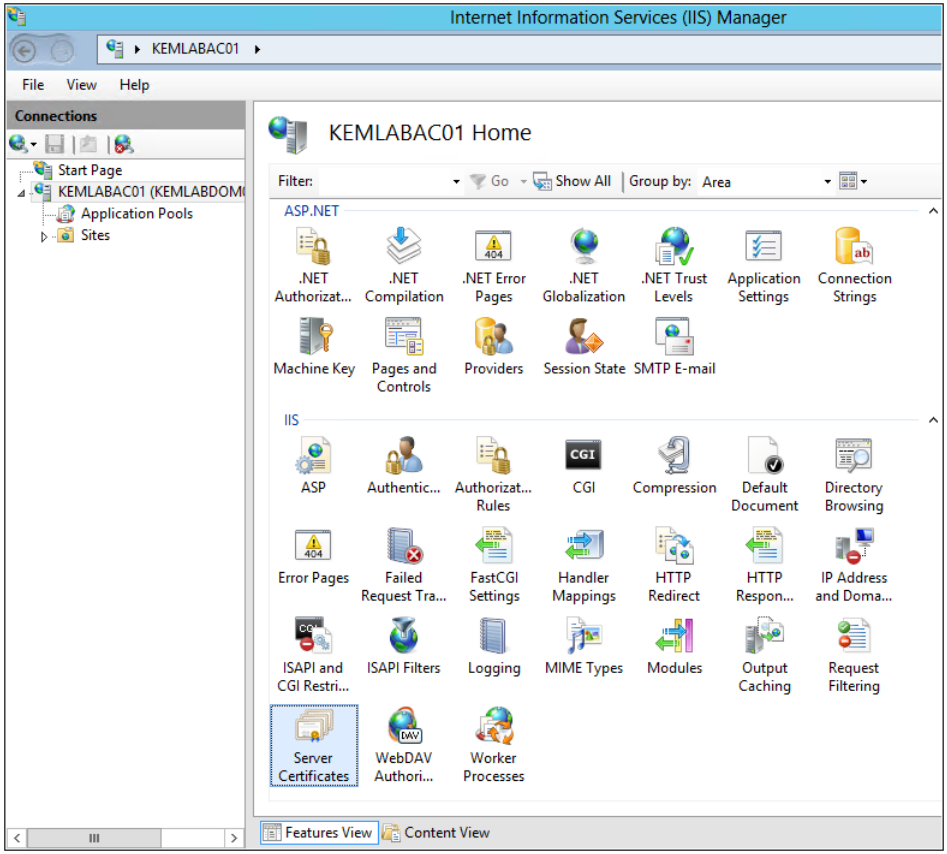


FIGURE 3-4 A view of the IIS Manager Home page in Features view.

On the Server Certificates page of the IIS Manager tool, you can then create a self-signed management certificate by clicking Create Self-Signed Certificate in the Actions pane located on the right side of the IIS Manager console as shown in Figure 3-5.

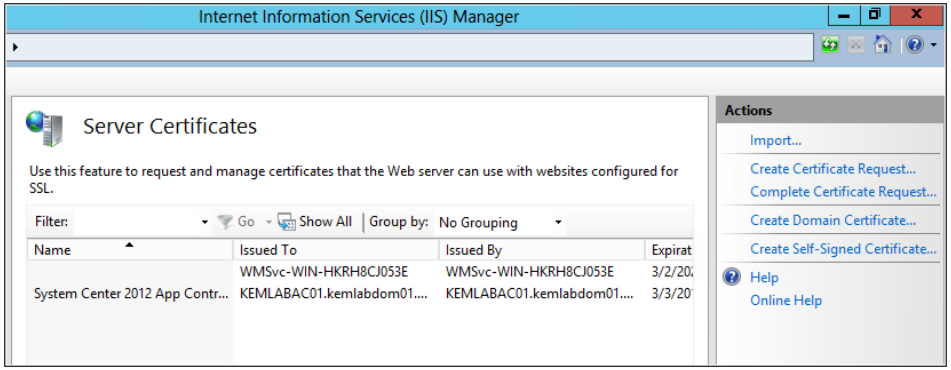


FIGURE 3-5 A view of the Server Certificates page in IIS Manager.

In the Create Self-Signed Certificate dialog box, enter a descriptive name for the new certificate being created and click OK to complete the certificate creation process, as shown in Figure 3-6.

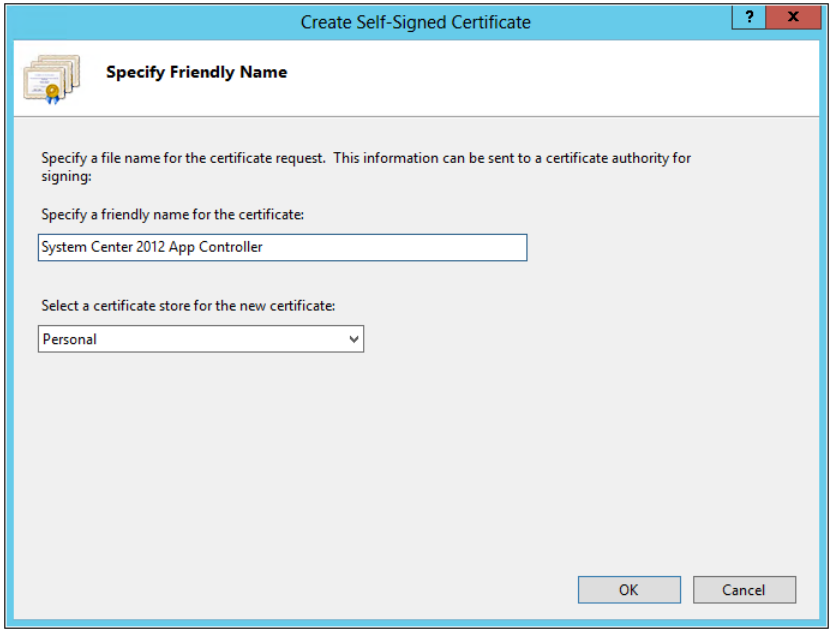

Create Self-Signed Certificate
Specify Friendly Name
Specify a file name for the certificate request. This information can be sent to a certificate authority for signing:
Specify a friendly name for the certificate:
System Center 2012 App Controller
Select a certificate store for the new certificate:
Personal
OK
Cancel

FIGURE 3-6 The Create Self-Signed Certificate dialog box allows you to name the certificate.

A new self-signed management certificate has now been created and should be visible on the Server Certificates page of the IIS Manager tool.

Uploading a management certificate to Windows Azure

After a management certificate has been created, the public keys associated with this certificate must be exported and uploaded to Windows Azure. This will provide Windows Azure with the credentials needed to authenticate your connection from App Controller. To export the public keys from the new certificate, right-click the certificate that is displayed on the Server Certificates page of the IIS Manager tool and select View from the pop-up context menu as shown in Figure 3-7.

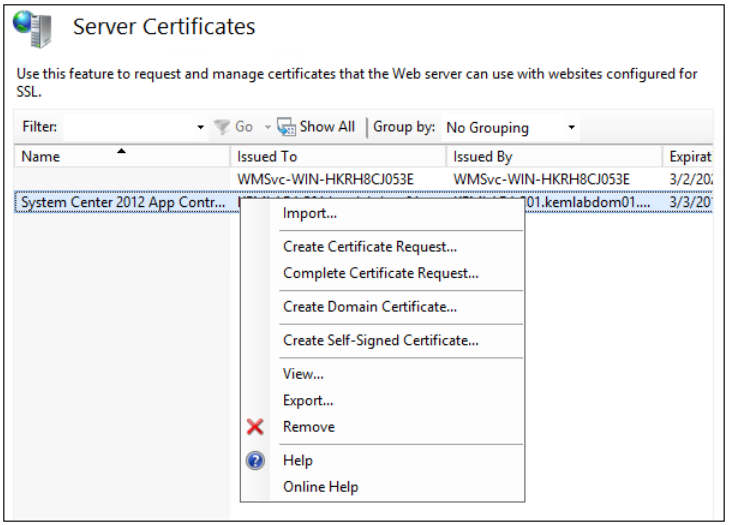


FIGURE 3-7 The context menu allows you to view an existing certificate.

Note that you should be careful to select the View option and not the Export option from the menu. Export saves a copy of both public keys and private keys associated with a certificate, and is useful for configuring the App Controller server, which you'll do in a later section of this chapter, or storing a local backup copy of the certificate. However, a certificate exported in this manner cannot be uploaded to the Windows Azure Management Portal because Windows Azure expects to receive only the public keys associated with a certificate.

After selecting View to show the new certificate properties, select the Details tab in the Certificate dialog box and click Copy To File, as shown in Figure 3-8.

In the Certificate Export Wizard, accept all default values and specify a filename to which you will export the public keys associated with your management certificate. After the export process is complete, you will have a DER-encoded binary X.509 certificate file with a .CER filename extension.

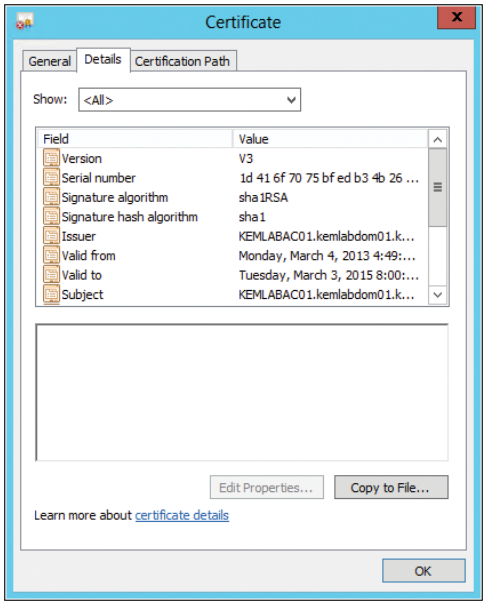


FIGURE 3-8 You can copy an existing certificate to a file from this dialog box.

To upload the exported certificate file to Windows Azure, navigate to the Windows Azure Management Portal at *http://manage.windowsazure.com* and sign in with the same Microsoft Account (that is, Windows Live ID) user credentials that you used when activating your Windows Azure subscription. On the Windows Azure Management Portal page, select Settings on the blue navigation bar located on the left side of the page (see Figure 3-9).

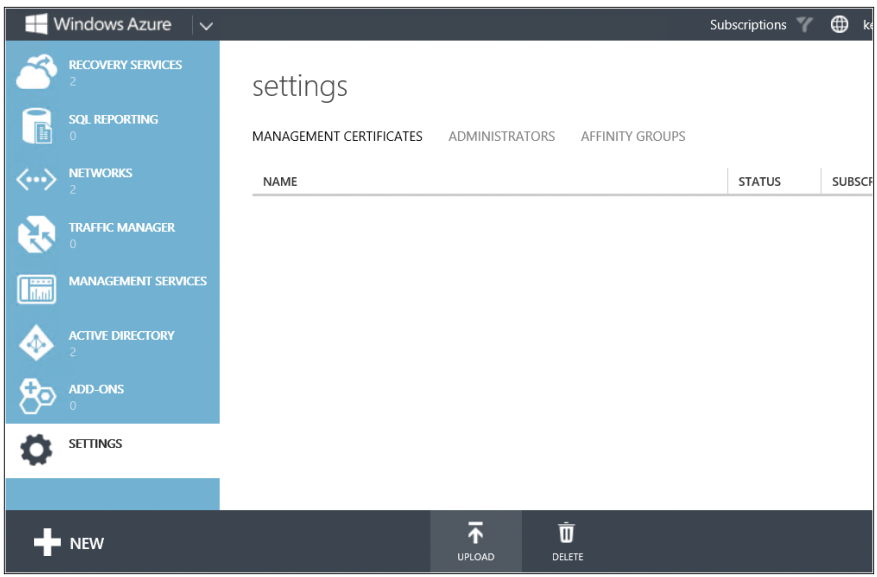


FIGURE 3-9 A view of the Settings page in the Windows Azure Management Portal.

On the Settings page, click the Management Certificates tab at the top of the page, and click Upload on the bottom black toolbar. In the Upload A Management Certificate dialog box, browse to the exported certificate file and select the name of your Windows Azure subscription. Click the checkmark button to upload the certificate file.

After the certificate file has been successfully uploaded, it will be listed on the Management Certificates tab in the Windows Azure Management Portal. Scroll to the right to display the Subscription ID associated with the Windows Azure subscription to which this certificate was uploaded (see Figure 3-10).

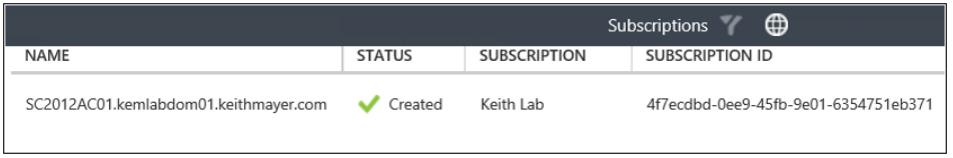


FIGURE 3-10 You can view the subscription ID in the Windows Azure Management Portal.

Make note of this Subscription ID before leaving this page. You'll need this ID value when establishing a new connection to this Windows Azure subscription from App Controller.

Connecting to public clouds

When connecting to a Windows Azure public cloud subscription, App Controller also needs to have a copy of the certificate being used to authenticate the connection. However, App Controller will need a copy of both the public keys and private keys associated with the certificate. This will permit App Controller to sign and authenticate the connection to Windows Azure.

To export a copy of the certificate with both public and private keys, you use the IIS Manager tool again. However, this time you use the Export option from the Server Certificates page to export a password-protected .PFX file that includes both public and private keys (see Figure 3-11).

When exporting the certificate to a .PFX file, specify a local path that will be accessible from the App Controller portal page. Remember the password that you enter to protect this exported file, because you'll need it when registering a new connection to your Windows Azure public cloud subscription.

Now that your management certificate has been exported to the App Controller server and you've uploaded the public keys to Windows Azure, you're ready to establish a new connection to your public cloud. To connect to a Windows Azure subscription from the App Controller portal, click Connect A Windows Azure Subscription on the Overview page of the App Controller portal. This link is located under Public Clouds on the page (see Figure 3-12).

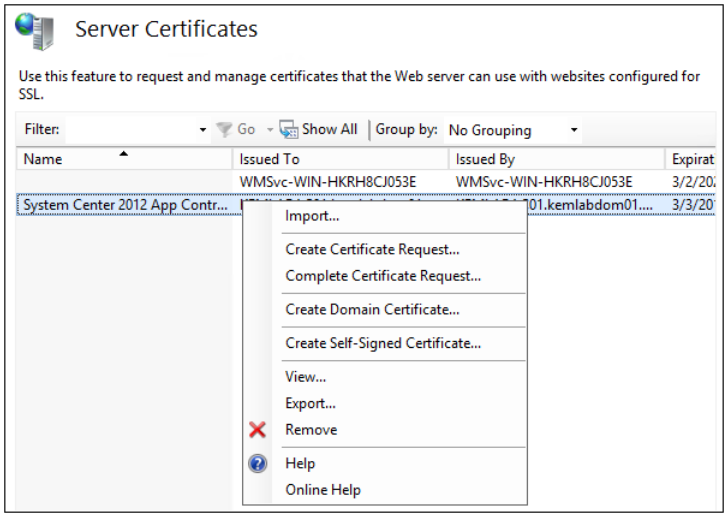


FIGURE 3-11 You can export a certificate using the IIS Manager tool.

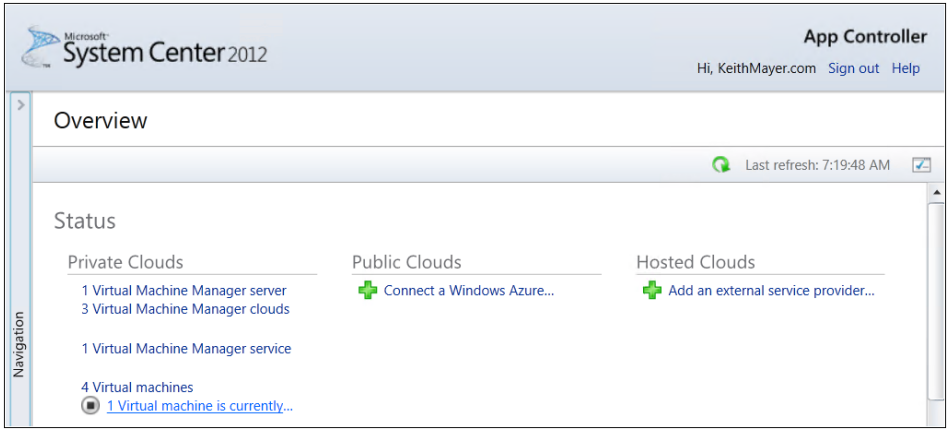


FIGURE 3-12 There are no public cloud subscriptions displayed in the App Controller Overview page.

In the Connect A Windows Azure Subscription dialog box shown in Figure 3-13, enter a name and description that can be used to identify this subscription from the App Controller portal. In addition, you need to enter the Subscription ID that you recorded from the Windows Azure Management Portal in the prior section of this chapter. Finally, you specify the path and password for the exported .PFX certificate file.

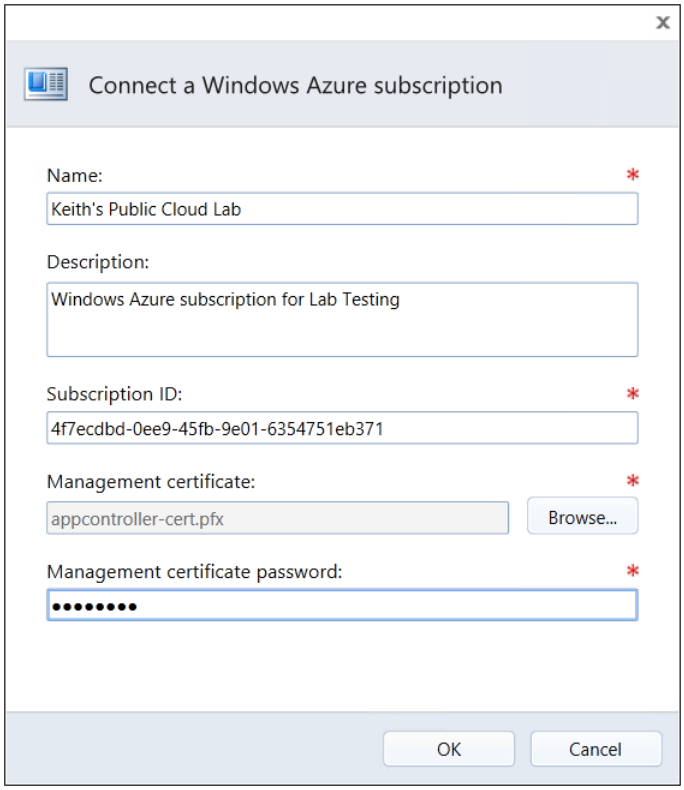


FIGURE 3-13 A view of the Connect A Windows Azure Subscription dialog box.

After the new connection has been successfully registered, the number of connected Windows Azure subscriptions will be displayed below Public Clouds on the Overview page as shown in Figure 3-14.

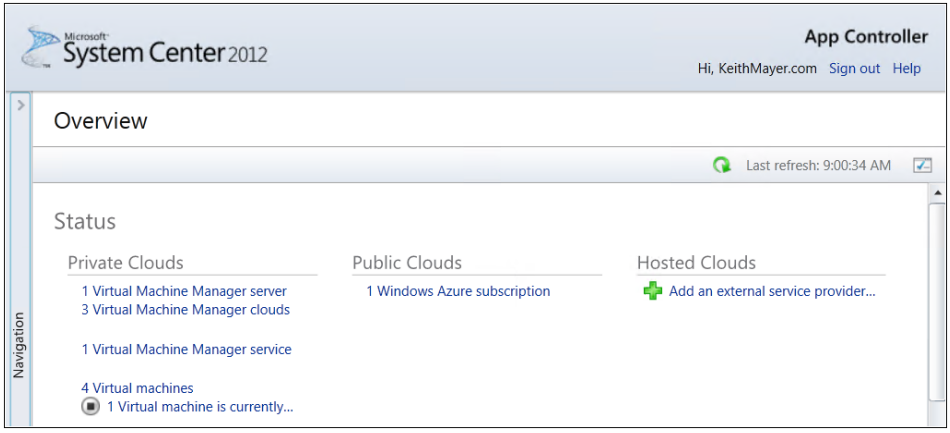


FIGURE 3-14 The Windows Azure subscription is listed as a connection under Public Clouds.

By clicking the link for 1 Windows Azure Subscription that is displayed below the Public Clouds on the Overview page, you will navigate to the Subscriptions page shown in Figure 3-15. From this page, you can add additional Windows Azure subscriptions, remove existing subscriptions, or view the properties of an existing subscription.

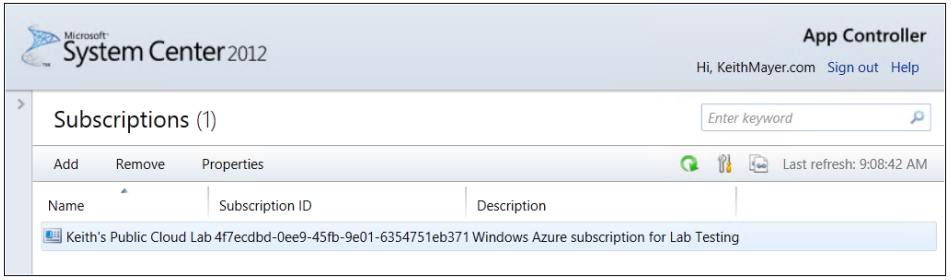


FIGURE 3-15 You can manage Windows Azure subscriptions from App Controller.

App Controller can support connections for up to 20 Windows Azure subscriptions per user from a single management portal. This can be useful when supporting centralized public cloud management in an organization where each business unit or team has been allocated their own subscriptions for charge-back or show-back purposes.

Delegating access to public clouds

After connections to Windows Azure subscriptions have been registered in the App Controller portal, you might want to delegate other Active Directory users or groups to have self-service access to one or more subscriptions. Delegation is useful in business scenarios where team members, other than the person owning the subscription, need access to provision and manage virtual machine workloads on Windows Azure. By providing these users with self-service access via the App Controller portal, these users will be able to leverage App Controller to provision and manage workloads across delegated Windows Azure subscriptions, without providing them with full administrative and billing access to these subscriptions.

To delegate one or more Active Directory users or groups with self-service access to Windows Azure, you can leverage the Settings, User Roles page in the App Controller portal as shown in Figure 3-16.

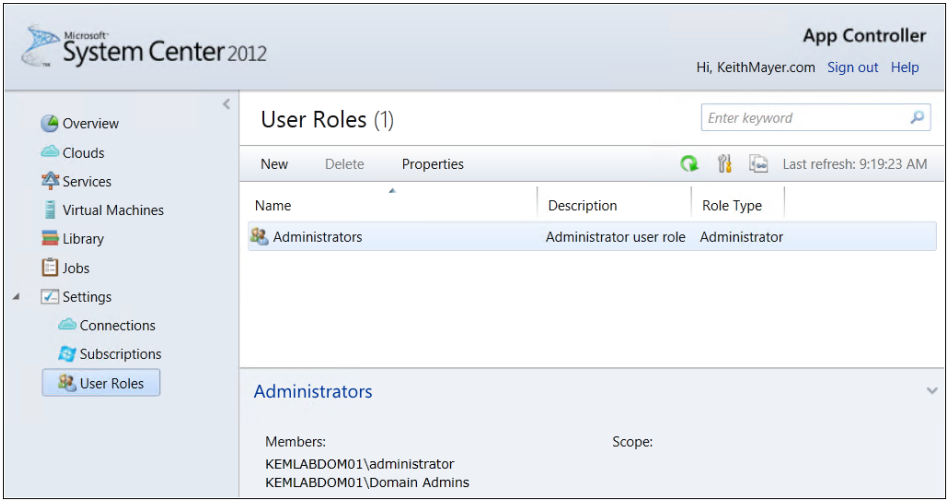


FIGURE 3-16 It is possible to manage user roles in App Controller.

On the User Roles page, click New on the top toolbar to begin delegating access to Windows Azure. In the New User Role dialog box shown in Figure 3-17, enter a name and description for the role that you are defining. If this role will only be used to view existing workloads that are already provisioned in Windows Azure, you can select the Read Only User Role check box. Otherwise, if this user role will be responsible for provisioning new workloads and managing existing workloads, leave this check box cleared. Under Members, click Add to add one or more Active Directory users and groups.

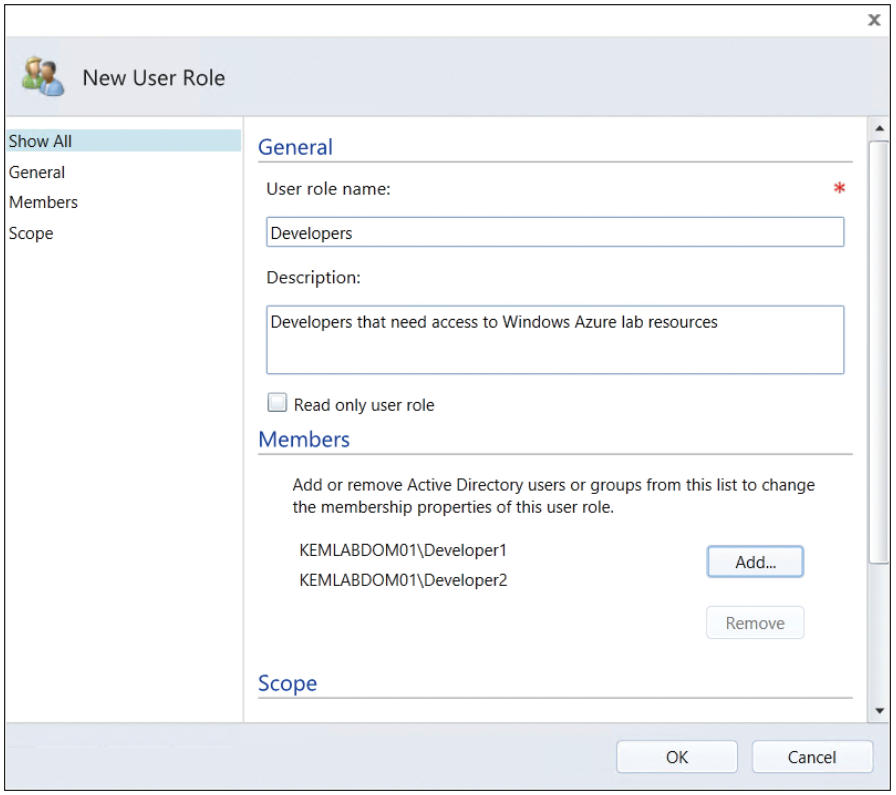


FIGURE 3-17 The New User Role dialog box allows you to add a role for delegating public cloud access.

When adding a new user role, you can also select a scope to limit the Windows Azure subscriptions to which the user role will have delegated access. Under Scope on the New User Role dialog box shown in Figure 3-18, select one or more Windows Azure subscriptions to which you will delegate self-service access.

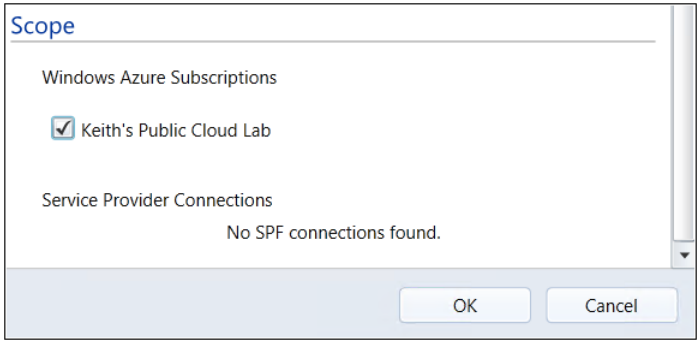


FIGURE 3-18 Select a Scope for delegated self-service access to Windows Azure subscriptions.

Click OK to add the new user role to App Controller.

Creating a Windows Azure storage account

Before deploying new virtual machine workloads to a Windows Azure public cloud subscription using App Controller, you will need to create a Windows Azure storage account. Storage accounts in Windows Azure provide a cloud-based storage location where virtual hard disks and other files can be stored. Storage accounts provide highly durable cloud-based storage by synchronously replicating all data across three separate physical disk locations in distinct upgrade domains and fault domains within the same Windows Azure data center. In addition, Storage accounts also provide asynchronous geo-replication between Windows Azure data centers. Any of these redundant data copies can be leveraged in the event of a physical disk or data center outage, providing an easy solution for recovery in the event of a disaster.

To create a new Windows Azure storage account from within the App Controller portal, navigate to the Library page as shown in Figure 3-19.

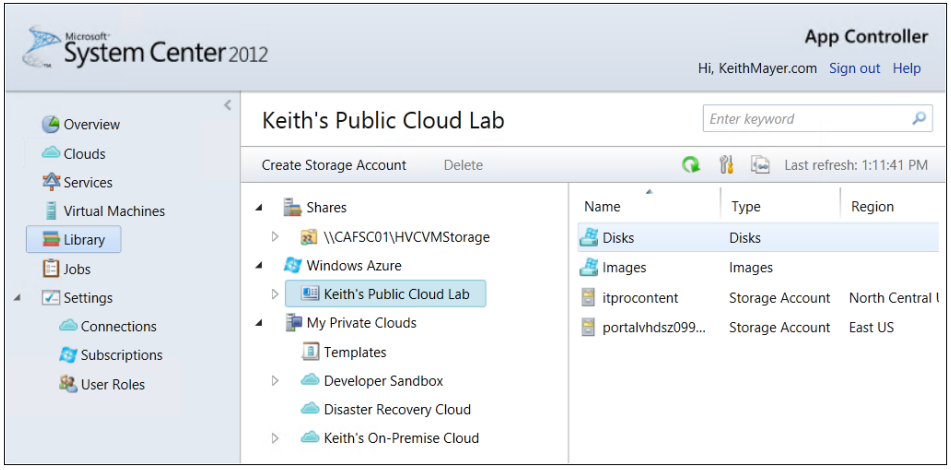


FIGURE 3-19 The Library page in the App Controller portal.

The Library page in the App Controller portal is used to manage storage operations, such as creating new storage accounts, copying, moving and deleting files, as well as adding virtual machine images and disks. To create a new storage account on the Library page, click the Windows Azure subscription that was previously connected and click Create Storage Account on the top toolbar. This will launch the Create A Windows Azure Storage Account dialog box as shown in Figure 3-20.

Using the Create A Windows Azure Storage Account dialog box, enter a unique name for the new storage account and select the Windows Azure data center region in which you'll be provisioning this new storage account. Be sure to select a Windows Azure data center region that is closest to your physical location to reduce network latency when accessing cloud services via the Internet.

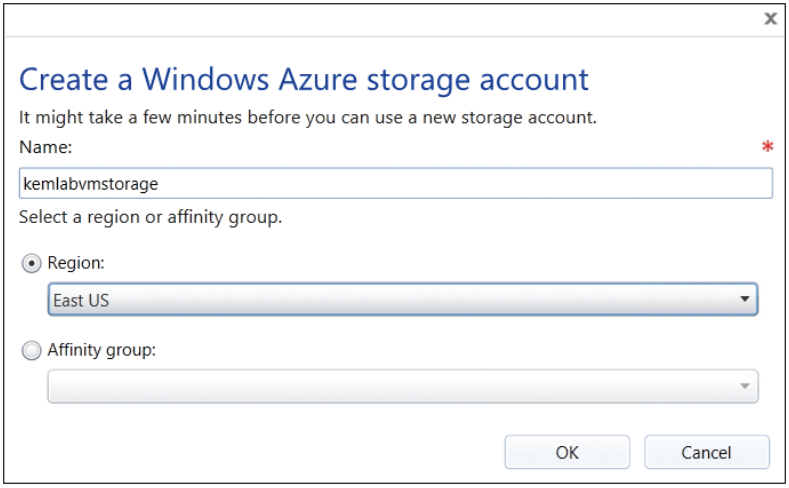


FIGURE 3-20 An example of the dialog box for creating a Windows Azure storage account.

When the information on the Create A Windows Azure Storage Account dialog box has been completed, click OK to provision a new storage account in your Windows Azure subscription. Provisioning a new storage account is processed as a background job via App Controller. To view the progress of this job, click the Jobs page within the App Controller portal as shown in Figure 3-21.

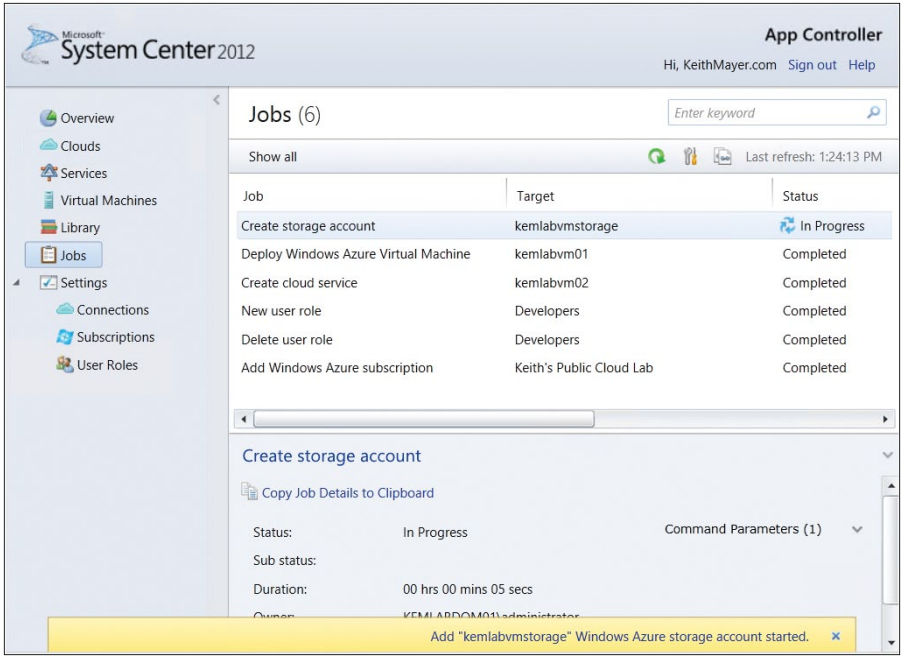


FIGURE 3-21 The yellow bar at the bottom allows you to monitor the creation of a Windows Azure storage account.

Once this job has been successfully processed, the Status column will be updated to reflect a Completed status for this job.

After a Windows Azure storage account is created, you need to create at least one container inside the new storage account in which to store virtual hard disks and files. A container in a Windows Azure storage account provides capabilities that are similar to a folder in an on-premises file system. To create a container, navigate back to the Library page in the App Controller portal and select the name of your newly provisioned storage account. Click Create Container on the top toolbar to create a new container name for storing virtual hard disk files, such as a container named **vhds** as shown in Figure 3-22.

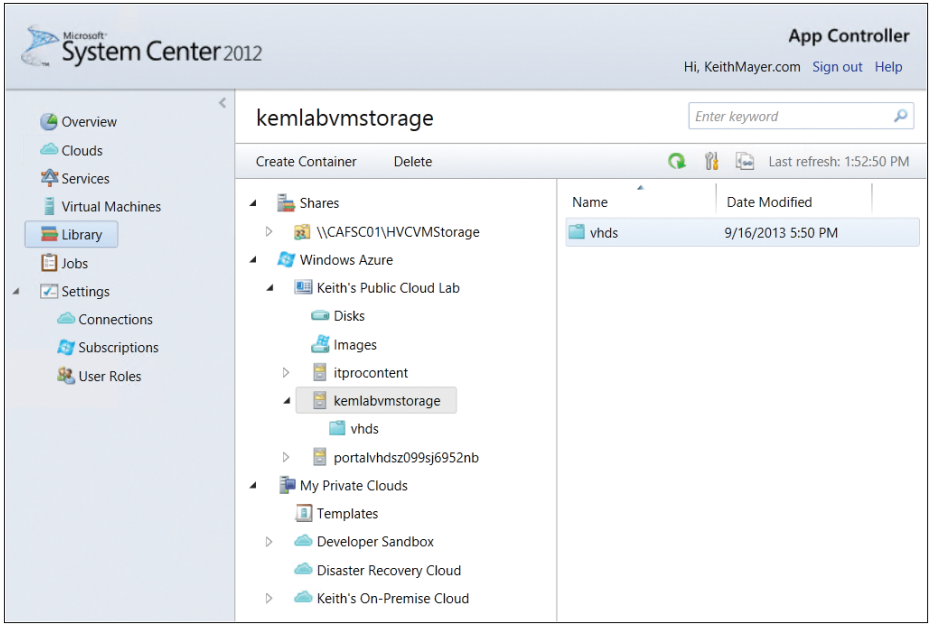


FIGURE 3-22 You can view a new container within a Windows Azure storage account.

Now that you have a new Windows Azure storage account and at least one container provisioned, you can move onward to deploying new virtual machine workloads in the Windows Azure public cloud.

Deploying new workloads to a public cloud

Using a similar process to what you saw in Chapter 2 in the context of private clouds, App Controller permits delegated users to deploy new virtual machine workloads to a Windows Azure public cloud subscription that is scoped within a role to which a user is assigned. To begin the process of deploying a new workload to Windows Azure, first select the Clouds page on the App Controller portal as shown in Figure 3-23. This page lists both private clouds and public clouds to which App Controller has previously been used to register cloud connections.

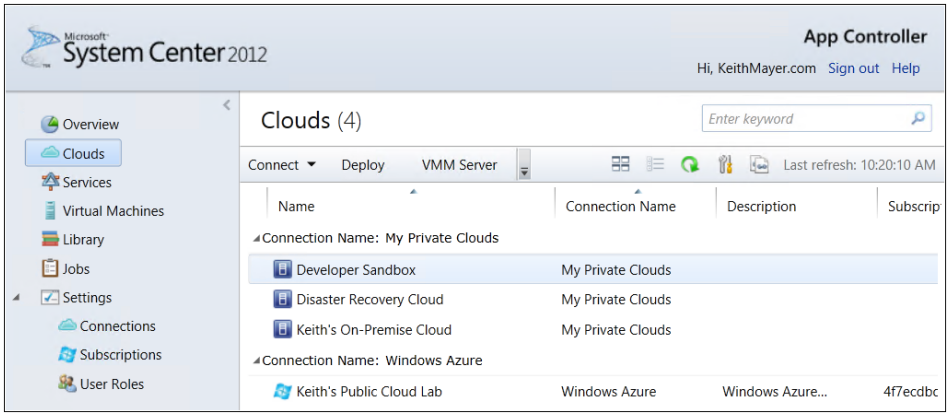


FIGURE 3-23 Developer Sandbox is selected on the Clouds page in the App Controller portal.

On the Clouds page, select one of the listed Windows Azure connections and click Deploy on the top toolbar to deploy a new virtual machine workload to that Windows Azure subscription. This will launch the New Deployment dialog box shown in Figure 3-24.

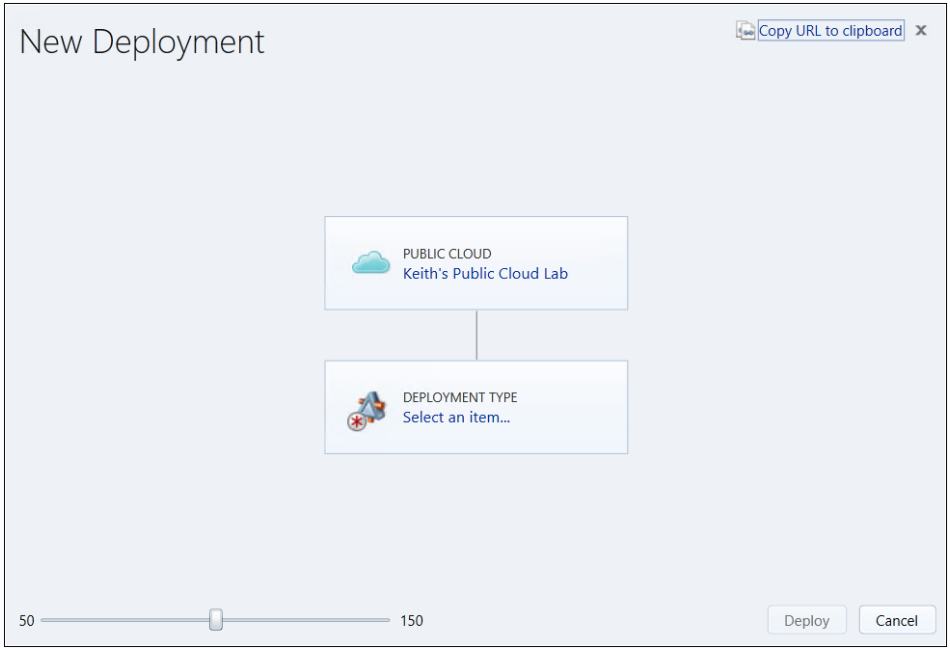


FIGURE 3-24 Keith's Public Cloud Lab is listed as deployed in the New Deployment dialog box.

In the New Deployment dialog box, click Select An Item located in the Deployment Type box to select an image to use for deploying a new virtual machine. This will launch the Select An Item To Deploy dialog box shown in Figure 3-25.

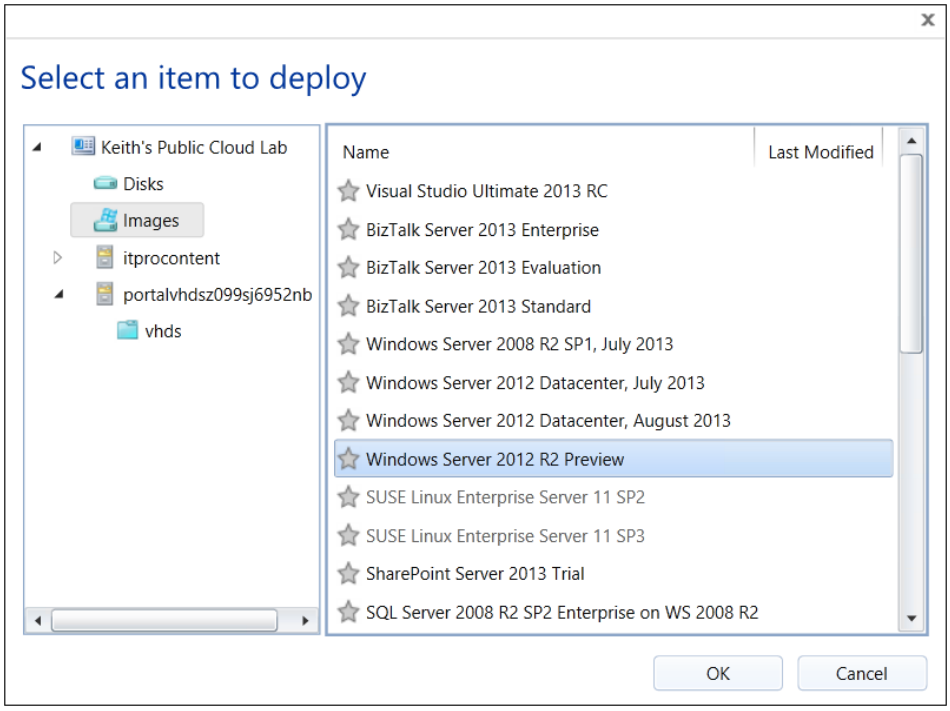


FIGURE 3-25 You select an image to use for deploying a new virtual machine.

Note that you have several locations from which you can select an image that can be used as the basis for deploying a new virtual machine:

- **Disks** These are virtual machine disks that you previously provisioned or uploaded and registered as a Disk in Windows Azure. Each disk can be attached to one virtual machine.
- **Images** Windows Azure includes a number of built-in platform images from which you can build one or more virtual machines. In addition, you can also select from any custom images that you've previously captured. Each image is generalized and can be used to provision one or more virtual machines.
- **Windows Azure Storage Accounts** In addition to selecting from Disks or Images that have previously been registered with Windows Azure, you can also browse through any storage accounts in the selected Windows Azure subscription for virtual hard disk files (that is, .VHD files) that have been previously uploaded to Windows Azure.

The process of capturing images and uploading VHD files to Windows Azure is discussed later in this chapter. For now, let's select one of the standard platform images as the basis for the new virtual machine that you're deploying. Click OK to select the platform image.

After you've selected the image to use as the basis for the new virtual machine, you'll be returned to the New Deployment dialog box, as shown in Figure 3-26, to complete the

configuration for the Cloud Service, Virtual Network, and Virtual Machine associated with this deployment.

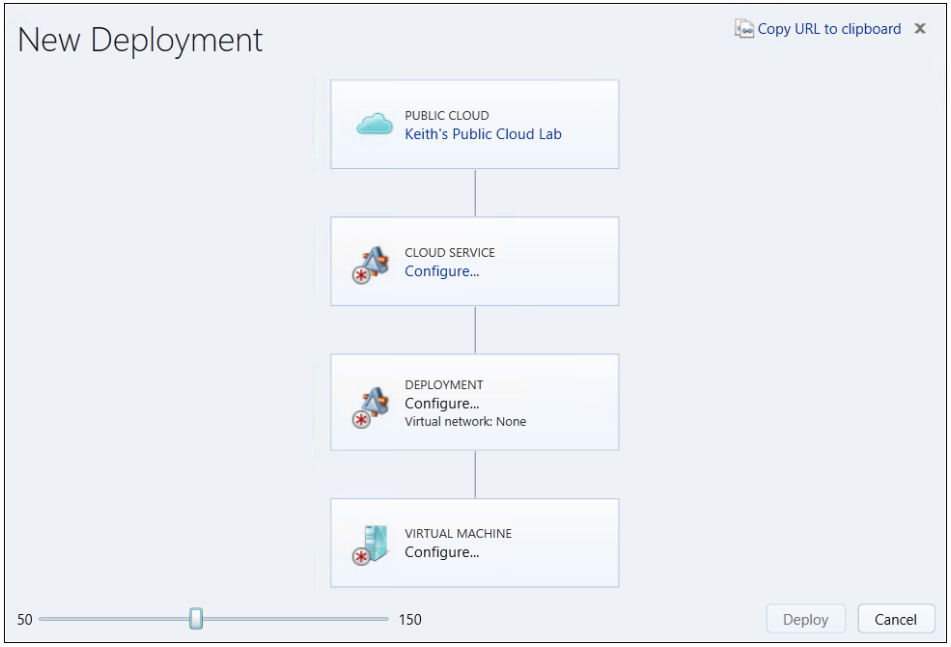


FIGURE 3-26 A new deployment with a cloud service, virtual network, and virtual machine.

You'll step through configuring each of these items over the next three subsections.

Configuring a cloud service

In Windows Azure, a cloud service provides a "container" in which one or more virtual machines can run. The cloud service provides several resources to the virtual machines contained within it—a public IPv4 address, a virtual firewall and load-balancer, and a public DNS hostname.

To configure a cloud service for this new deployment, click Configure inside the Cloud Service box in the New Deployment dialog box. This will launch the Select A Cloud Service For This Deployment dialog box, where you can either select a pre-existing cloud service or create a new cloud service for this deployment. Click Create to create a new cloud service. This will launch the Create A Cloud Service dialog box.

In the Create A Cloud Service dialog box shown in Figure 3-27, enter a name and description for this cloud service. You also need to enter a unique public DNS hostname for this cloud service. Finally, in the Deploy To A Specific Region list box, select a Windows Azure data center region to which this cloud service should be deployed. Select the data center region that is geographically closest to you, so that network latency will be lower when accessing your cloud service over the Internet.

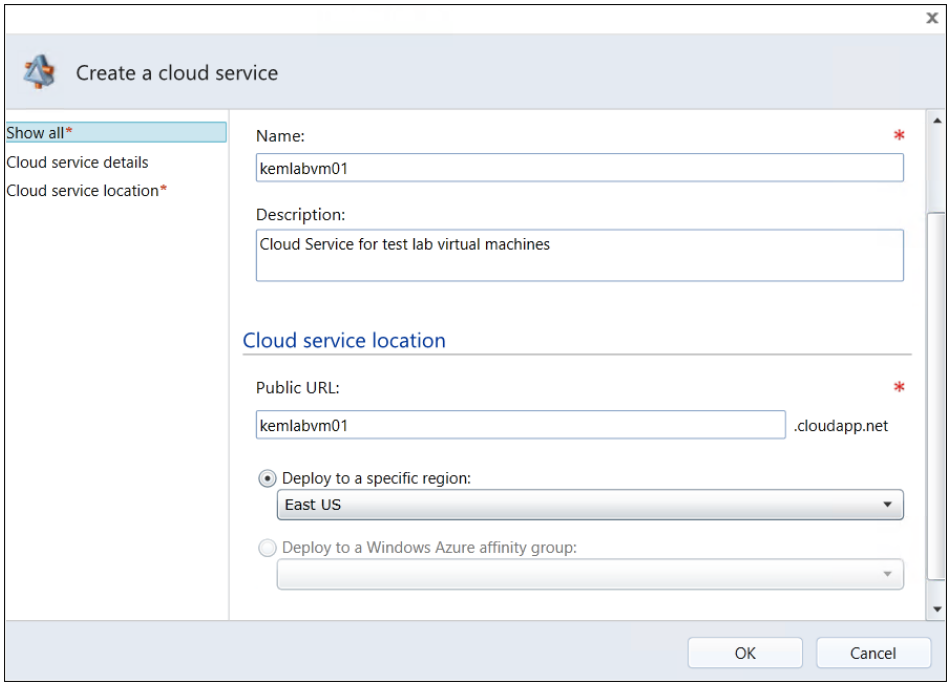


FIGURE 3-27 An example of creating a new Windows Azure cloud service.

When you've finished entering the information in the Create A Cloud Service dialog box, click OK to create a new cloud service. In the list of cloud services, select the cloud service you just created and click OK to be returned to the New Deployment dialog box.

Configuring a virtual network

Windows Azure uses virtual networks to define a private IPv4 address space for virtual machines that are deployed in one or more cloud services. By using a virtual network, a specific IP address range can be configured for virtual machines without a virtual network in place. Windows Azure will dynamically assign private IP addresses of its own choosing for each virtual machine. Virtual networks can also host site-to-site and point-to-site virtual private networks (VPN) for securely connecting a Windows Azure Virtual Network to on-premises VPN gateways or remote client devices via VPN tunnels. Windows Azure Virtual Network is a detailed topic that is outside the scope of this book, but to learn more about building virtual networks you can leverage the *Build Virtual Networks in Windows Azure Step-by-Step Lab Guide* that is available at *http://aka.ms/VNetCloudLab*.

You don't currently have a virtual network defined in Windows Azure, so click Configure inside the Deployment box on the New Deployment dialog box and just specify a Name for the deployment as shown in Figure 3-28. Leave the Virtual Network set to None. In this case, Windows Azure will dynamically assign a private IPv4 address to the new virtual machine being deployed, because you have selected to not use a virtual network.

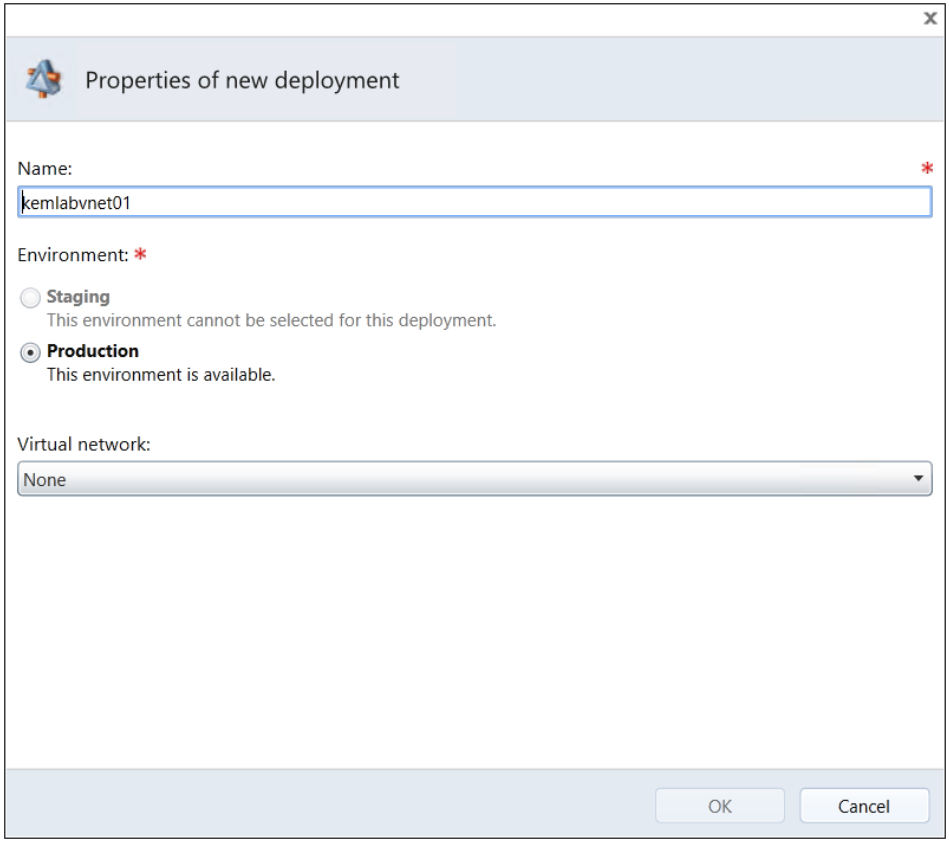


FIGURE 3-28 The Properties Of New Deployment dialog box allows you to set configurations for the new deployment.

Click OK when you've completed the Name text box on the Properties Of New Deployment dialog box. You will be returned to the New Deployment dialog box.

Configuring a virtual machine

The last item to configure for a new deployment involves the properties of the new virtual machine being deployed. Click Configure inside the Virtual Machine box on the New Deployment dialog box (Figure 3-26) to specify the desired properties of the new virtual machine as shown in Figure 3-29.

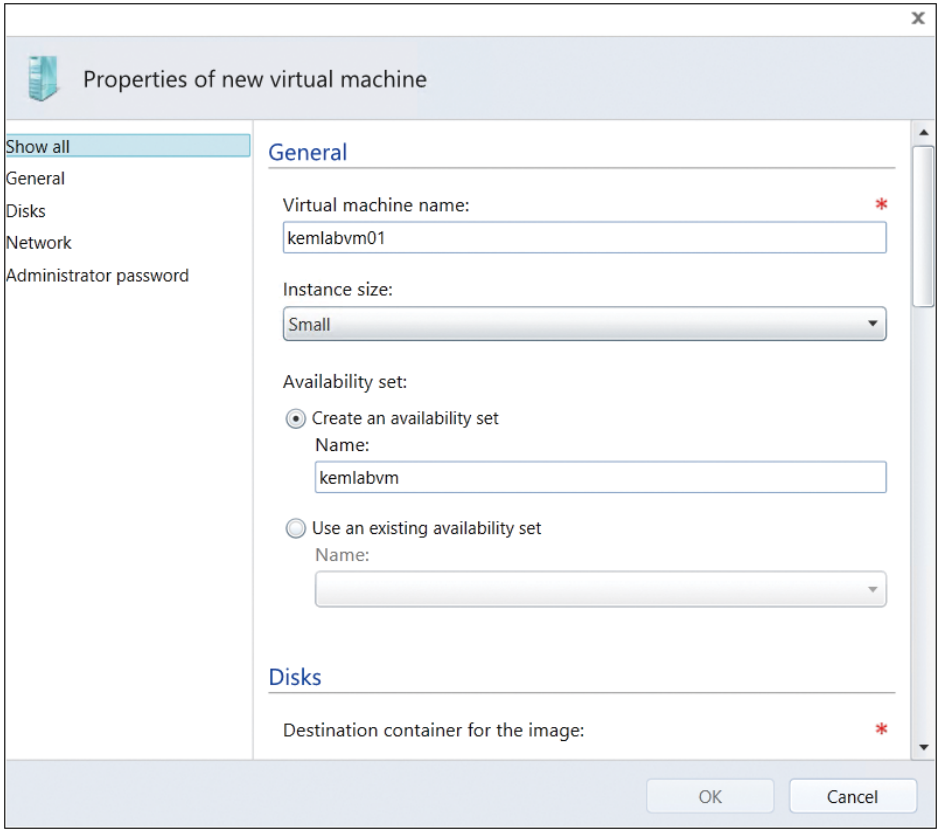


FIGURE 3-29 You can configure the properties in the Properties Of New Virtual Machine dialog box.

In the Properties Of New Virtual Machine dialog box shown in Figure 3-29, specify the following values:

- **Virtual Machine Name** Enter a unique name that will be provisioned as the computer name or hostname for the new virtual machine.
- **Instance Size** Specify the size of the new virtual machine to allocate a specific amount of processor and memory resources. Details on virtual machine sizes and associated costs can be found on the Windows Azure pricing page located at *http://www.windowsazure.com/en-us/pricing/calculator/?scenario=virtual-machines*.
- **Availability Set** Create a new availability set name for this first virtual machine. Availability set names in Windows Azure are used to identify multiple virtual machines that are running a common application workload within a cloud service. By tagging multiple virtual machines with the same availability set name, Windows Azure automatically locates these virtual machines in separate upgrade domains and fault domains to provide the highest possible level of application availability during planned and unplanned downtime windows. Frequently, availability sets are used when load-balancing an application across several virtual machines.

- **Disks** Click Browse to browse to a container within an existing Windows Azure storage account in which the virtual hard disk files associated with this new virtual machine should be stored. Prior to deploying a new virtual machine, you must create at least one Windows Azure storage account using the Windows Azure Management Portal. In addition to specifying the Storage Account path for storing virtual hard disks, you may also optionally click Add to add one or more existing virtual hard disks to this new virtual machine for storing data files, or click Create to create new blank virtual hard disks for storing data files.
- **Network** Use the Add and Remove buttons to configure one or more network endpoints in the Windows Azure virtual firewall for this new deployment. These endpoints will selectively permit inbound network traffic to the public IPv4 address of the cloud service via the public port defined for each endpoint. When traffic is received on the public port, it is forwarded to the private port on the private IP address of the associated virtual machine. In the example displayed in Figure 3-30, we've added an additional firewall endpoint for HTTP web traffic to listen on TCP port 80.

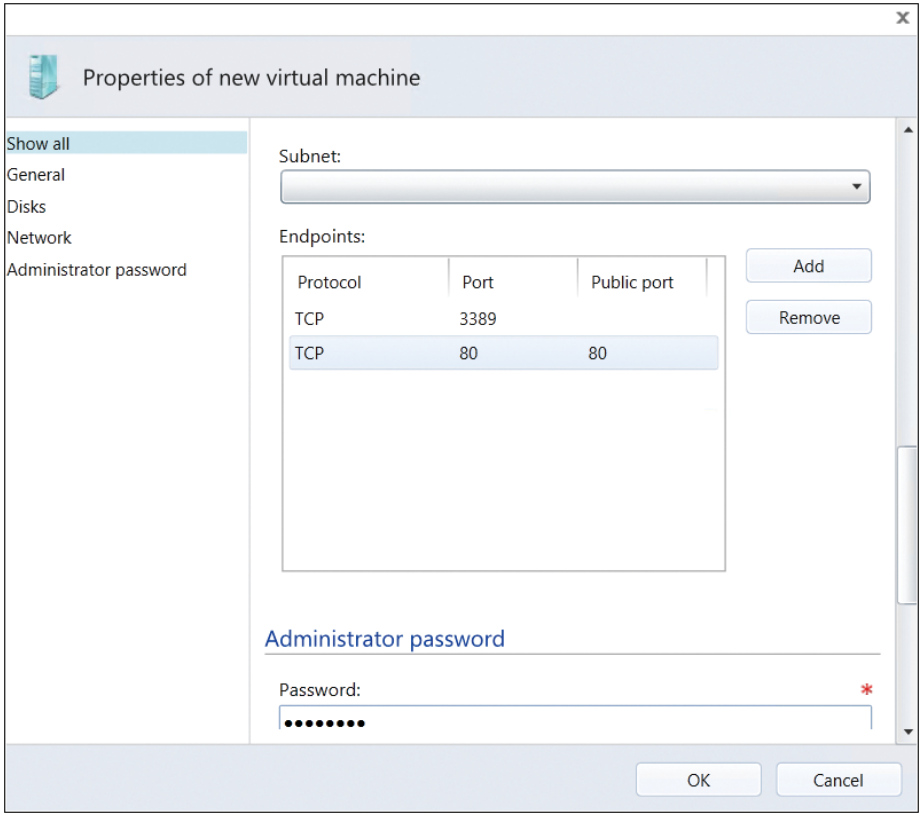


FIGURE 3-30 You can configure firewall endpoints and the local administrator password for new the virtual machine.

Note that configuring endpoints only configures the Windows Azure virtual firewall. If the operating system running within a virtual machine also includes a firewall, such as Windows Firewall with Advanced Services, you will need to confirm that this firewall permits the allowed inbound traffic as well.

If using a Windows Azure Virtual Network, you can also specify a particular subnet within the virtual network for placing this new virtual machine. Since you're not using a virtual network for this deployment, leave the Subnet text box blank.

- **Administrator Password** Enter and confirm a password to be provisioned for the local Administrator account when this new virtual machine is deployed. If this new virtual machine is being placed on an existing virtual network where Windows Server Active Directory is already present, you can also optionally specify an Active Directory domain name and user credentials to join this new virtual machine to Active Directory as part of the deployment process.

When all the information is complete on the Properties Of New Virtual Machine dialog box, click OK to save these property values. You will be returned to the New Deployment dialog box as shown in Figure 3-31.

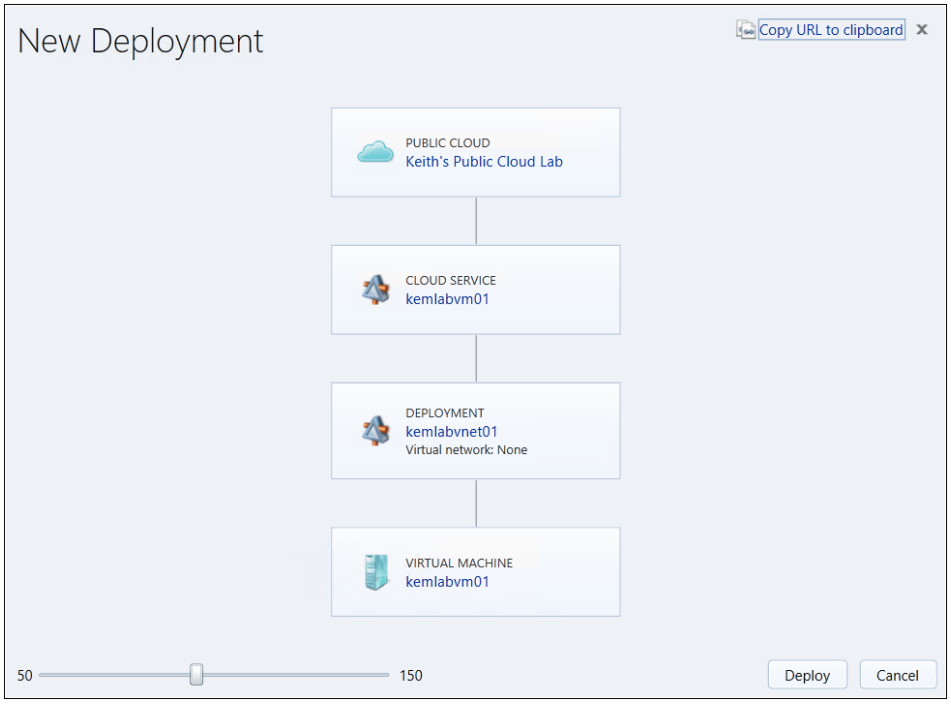


FIGURE 3-31 A view of the updated details for the deployment.

Ready to deploy

When all boxes on the New Deployment dialog box have been completed, click Deploy to begin your new deployment to the Windows Azure public cloud subscription. As the new deployment is processed, you can monitor progress on the Jobs page of the App Controller portal (see Figure 3-32).

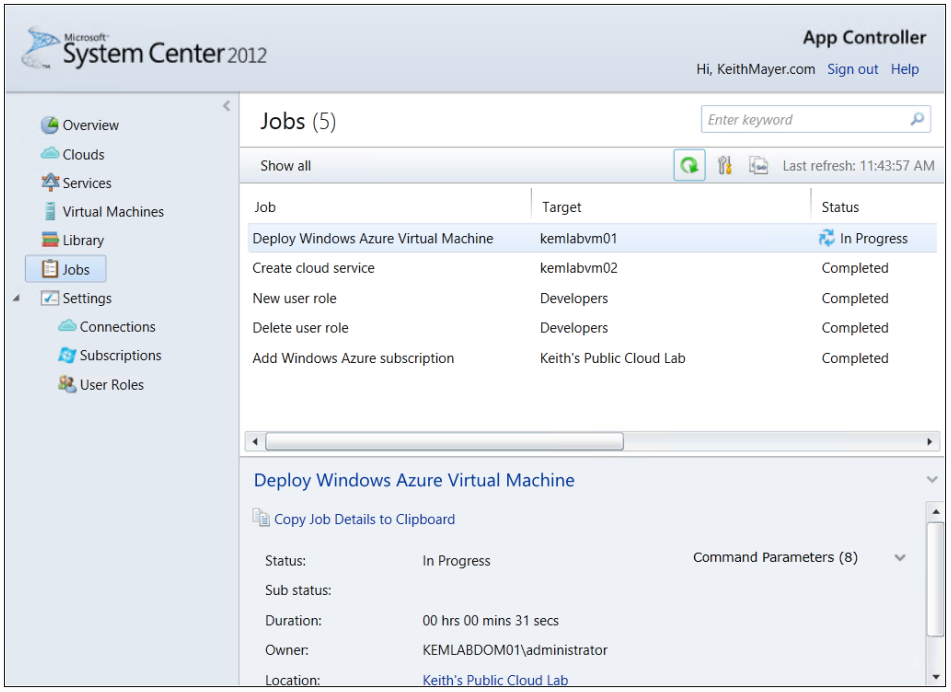


FIGURE 3-32 The Jobs page allows you to monitor deployment progress.

The deployment process will take a few minutes to complete. When successfully completed, the Jobs page will list the job status for the deployment job as Completed.

After deploying your first virtual machine within a cloud service, additional virtual machines can be easily deployed to the same cloud service or virtual network simply by selecting an existing cloud service or virtual network name when deploying new virtual machine workloads (see Figure 3-33).

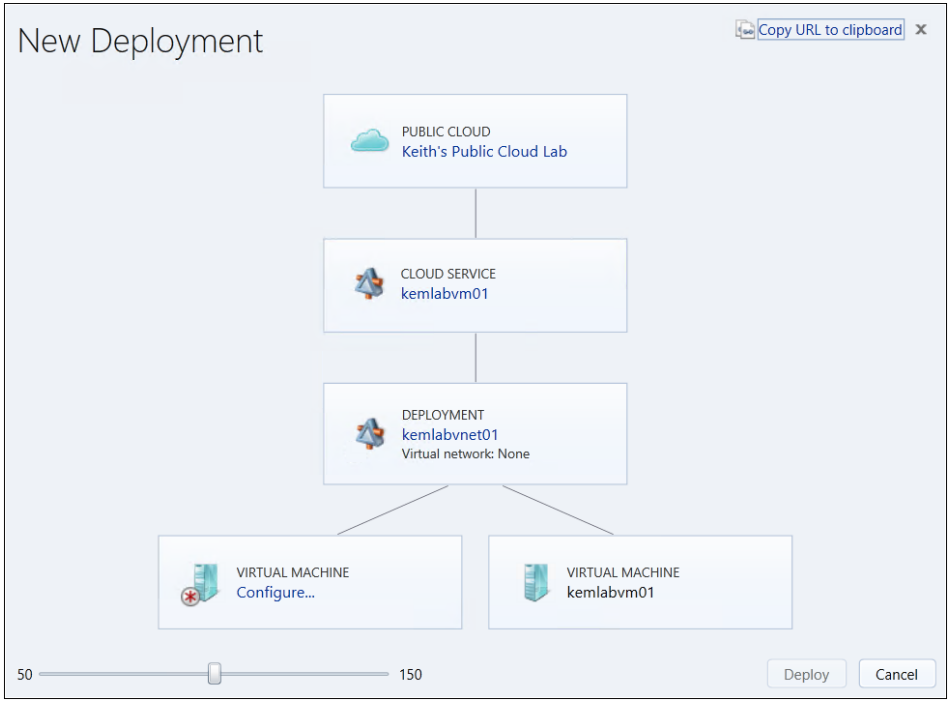


FIGURE 3-33 You can deploy a second virtual machine to an existing cloud service.

This type of multi-virtual machine deployment is commonly performed when load-balancing multiple virtual machines that are running common application workloads, or when deploying multi-tier applications on a common virtual network. Using the visual New Deployment user interface, App Controller provides an easy method for provisioning simple single-VM application workloads as well as complex workloads consisting of more than one virtual machine.

Managing public cloud workloads

After being provisioned, public cloud workloads on Windows Azure can also be managed via the App Controller portal. Two portal pages are available for managing public cloud workloads:

- **Services page** Permits the ability to monitor, stop, and delete cloud services as an entity. If a cloud service includes more than one virtual machine, these operations will be effective for all virtual machines within the selected cloud service.
- **Virtual Machines page** Permits the ability to manage an individual virtual machine with actions such as monitor, modify properties, delete, shutdown, restart, and establish a remote desktop connection.

When taking a serviced-oriented approach to managing public cloud workloads, navigate to the Services page in the App Controller portal. By right-clicking a deployed cloud service, you have access to the operations that can be performed on that cloud service and all virtual machines within it (see Figure 3-34).

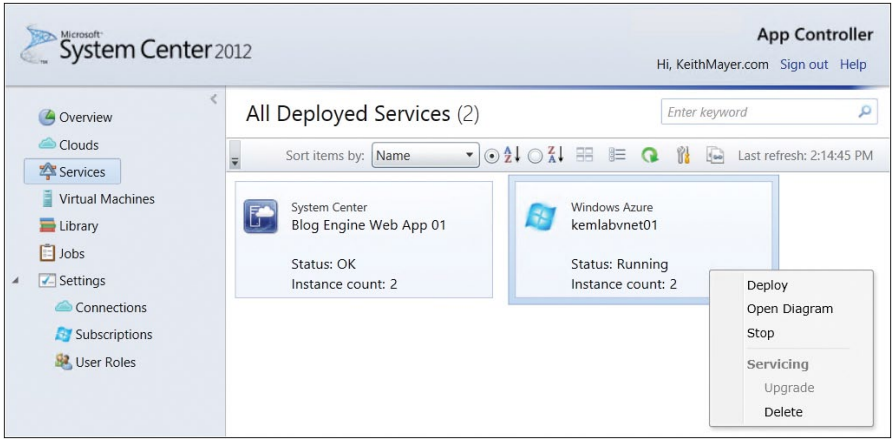


FIGURE 3-34 Use the Services page to manage cloud services deployed to Windows Azure.

When you're managing individual virtual machines within a cloud service, navigate to the Virtual Machines page in the App Controller portal. By right-clicking a deployed virtual machine, you have access to the operations that can be performed on that single virtual machine only (see Figure 3-35).

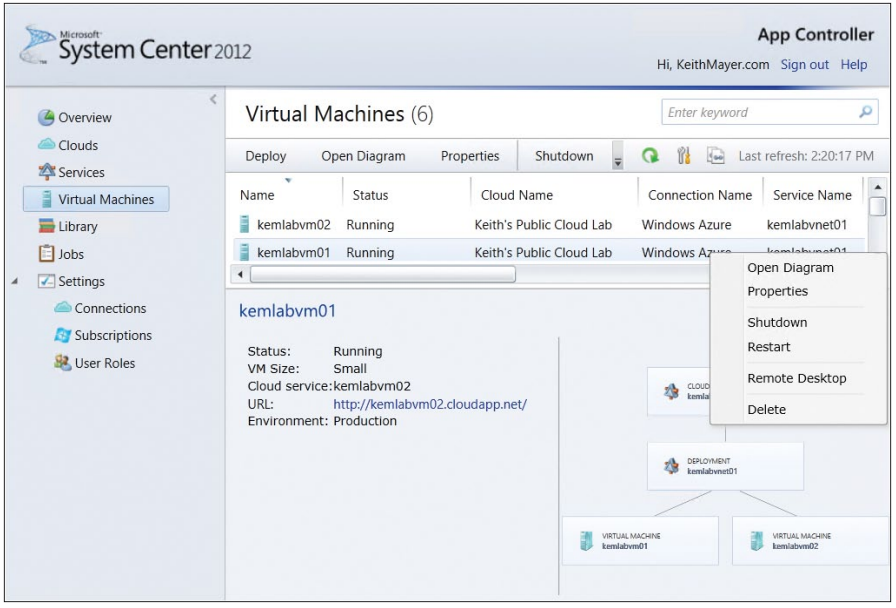


FIGURE 3-35 You can manage individual virtual machines deployed to Windows Azure.

Note that when stopping cloud services or performing a shutdown of virtual machines from the App Controller portal, the configuration, processor, memory, and IP address resources are kept in a reserved state. Because these reservations are maintained, Windows Azure continues to accumulate compute charges for cloud services and virtual machines in this state. If you want to stop cloud services or virtual machines and de-allocate these resource reservations to avoid compute charges while in a stopped state, you should consider stopping cloud services and virtual machines from the Windows Azure Management Portal instead. The Windows Azure Management Portal can be accessed via a web browser at *http://manage.windowsazure.com*.

Managing files, disks, and images in public clouds

When managing public clouds from the App Controller portal, certain tasks may require managing files between network file shares and public cloud storage accounts. Virtual machines on Windows Azure use the same virtual hard disk format as virtual machines running on an on-premises Windows Server 2012 Hyper-V host, and this provides a great deal of portability when managing cloud workloads. For instance, you might already have a set of virtual machine images that you want to move to a public cloud for provisioning in Windows Azure. Alternatively, you might have one or more virtual machines that are already provisioned in a public cloud using Windows Azure, and you might want to move the virtual hard disks associated with those VMs to an on-premises network file share for provisioning locally on a Hyper-V host. Both tasks can be easily accomplished using the App Controller portal.

Moving files to/from Windows Azure storage accounts

Moving files between network file shares and public cloud storage accounts can be performed via the Library page in the App Controller portal, as shown in Figure 3-36.

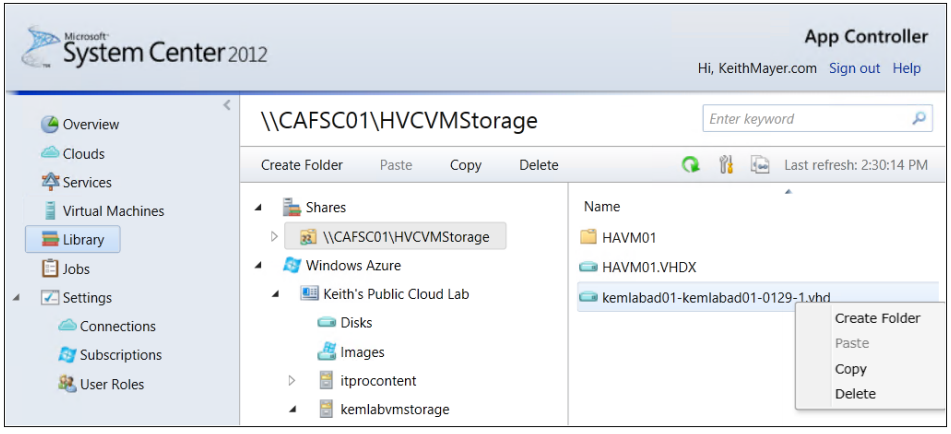


FIGURE 3-36 Right-clicking a virtual hard disk file displays a context menu that allows you to move files.

To move files, such as virtual hard disks (for example, .VHD files), from a network file share to a public cloud storage account, select the files from the network file share that was previously added to the App Controller portal in Chapter 2 and right-click. On the right-click menu, select Copy. After the files have been selected for a copy operation, click the appropriate public cloud storage account and container. Click Paste on the top toolbar to begin the copy operation.

To move files from a public cloud storage account to a network file share, simply reverse this process by selecting the files for a Copy operation from within the appropriate storage account and container, and then Paste into the desired network file share.

Note that for virtual hard disk files to be portable between Windows Azure and an on-premises deployment of Windows Server 2012 Hyper-V, the following considerations must be kept in mind:

- Windows Azure currently supports only fixed-size VHD files. Dynamic VHD files are not supported.
- Windows Azure currently supports VHD files up to 1 TB in size.
- Windows Azure currently supports only virtual hard disk files in the VHD format. The new VHDX virtual hard disk format introduced by Windows Server 2012 Hyper-V is not supported on Windows Azure. If you are using other virtual hard disk formats in your on-premises data center, they must first be converted to a fixed VHD file before uploading to Windows Azure.
- When uploading VHD files that include a bootable operating system, the DHCP client service must be configured to automatically start and the network adapter must be configured to receive an IP address via DHCP. Static IP addresses are not currently supported on Windows Azure virtual networks.
- Remote desktop management, or Secure Shell (SSH) for Linux operating systems, must be enabled on VHD files that include a bootable operating system so that remote console administration is accessible after a new virtual machine is deployed using that VHD file.

Adding disks and images

When attaching VHD files to Windows Azure virtual machines, they must first be registered as either a disk or an image in your Windows Azure subscription. By registering as a disk or an image, Windows Azure will create a lease on the underlying VHD file that will protect it from accidental deletion. The process for registering disks and images is very similar, but disks and images each serve different purposes:

- **Disks** Register an uploaded VHD file as a disk on Windows Azure when using that VHD file as an operating system disk or data disk for one virtual machine. Disks are normally used to register uploaded VHD files when migrating an existing on-premises virtual machine to Windows Azure.

- **Images** Register an uploaded VHD file as an image on Windows Azure when using that VHD file as a generalized operating system image for provisioning multiple virtual machines. Images can be registered for generalized VHD files that contain an operating system image that has been prepared for imaging, often with Windows imaging preparation tools such as Sysprep.

Use the Library page in the App Controller portal to manage disks and images for your Windows Azure subscription. From the Library page, select your Windows Azure connection and then click either the Disks or Images folder below it, depending on which type of object you want to manage (see Figure 3-37).

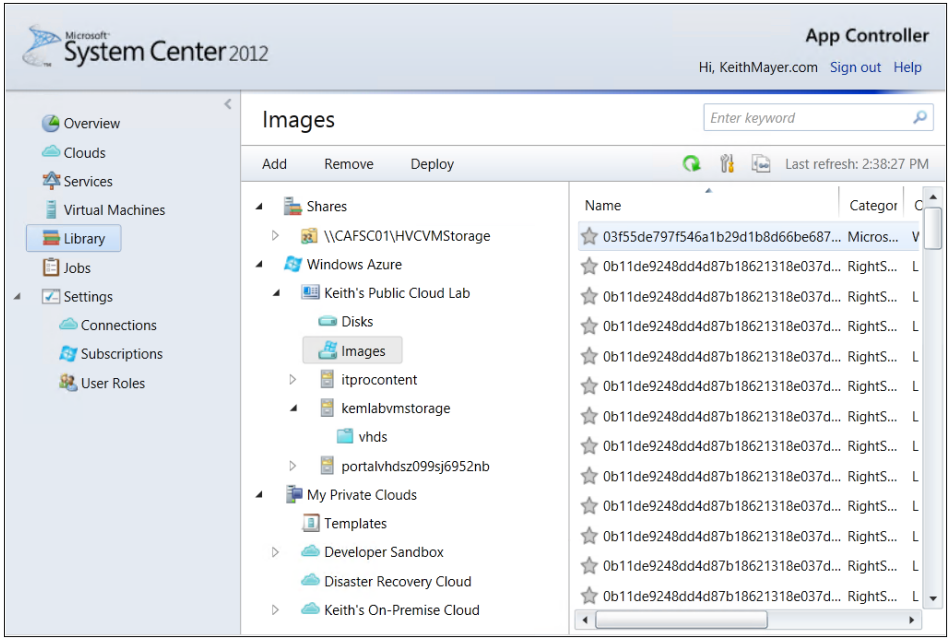


FIGURE 3-37 You can manage disks and images from the Library page.

After selecting either the Disks or Images folder on the Library page, use the top toolbar to Add and Remove disks or images, as well as Deploy new virtual machines based on that disk or image.

CHAPTER 4

Managing hybrid clouds

This chapter examines using Microsoft System Center 2012 R2 App Controller to manage hybrid computing environments that combine together System Center 2012 R2 Virtual Machine Manager (VMM) private clouds and the Windows Azure public cloud. Managing such a hybrid environment is easy with App Controller because it provides a single interface that can be used for connecting to and managing cloud resources either on- or off-premises.

Because we've already covered using App Controller to separately manage private and public cloud environments in the previous two chapters, we won't repeat what was covered previously and instead will focus on the following tasks which are commonly performed in hybrid environments:

- Copying a virtual hard disks (VHD) from VMM to Windows Azure
- Deploying a cloud service in Windows Azure using an uploaded VHD
- Copying virtual machines from VMM to Windows Azure

MORE INFO **To effectively use App Controller user managed resources deployed in a hybrid computing environment, it is essential to understand the VMM private cloud deployment model using service templates and the Windows Azure Infrastructure as a Service (IaaS) deployment methodology. You might find the following resources helpful in this regard:**

- *http://aka.ms/SCVMM*
- *http://aka.ms/AzureIaaSMethod*

Copying a VHD from VMM to Windows Azure

A basic need in a hybrid environment is the ability to easily move virtual hard disks (VHDs) from an on-premises VMM-based private cloud to the Windows Azure public cloud. Possible reasons for doing this include offsite storage, ad hoc deployment, on-demand testing/troubleshooting, or piloting a deployment. App Controller makes it simple to perform this task using copy-and-paste as you will see in this walkthrough.

You begin by logging on to App Controller and selecting the Library workspace, which displays all the resources the logged-on user is authorized to use such as shares

added by VMM library servers, disks, images, and storage accounts from connected Windows Azure subscriptions. The highlighted VHD file in Figure 4-1 is a sysprepped image in a shared folder in the private cloud managed by VMM. Right-clicking the VHD file displays a shortcut menu with the option for copying the VHD as shown in the Figure 4-1.

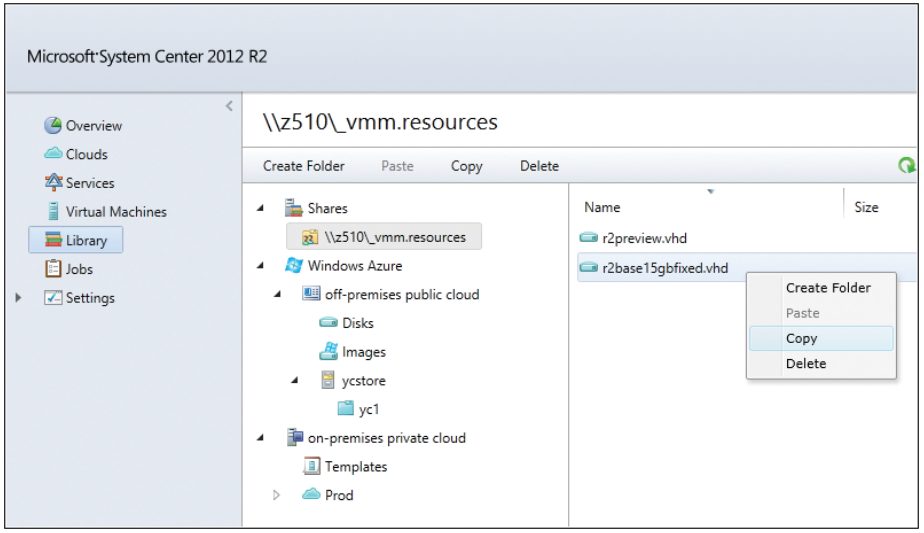


FIGURE 4-1 A VHD can be copied from a shared folder using App Controller.

Once copied, you can then paste the VHD into a storage container in the connected Windows Azure subscription. You can do this by selecting the container and then right-clicking in the rightmost pane and selecting the Paste option as shown in Figure 4-2.

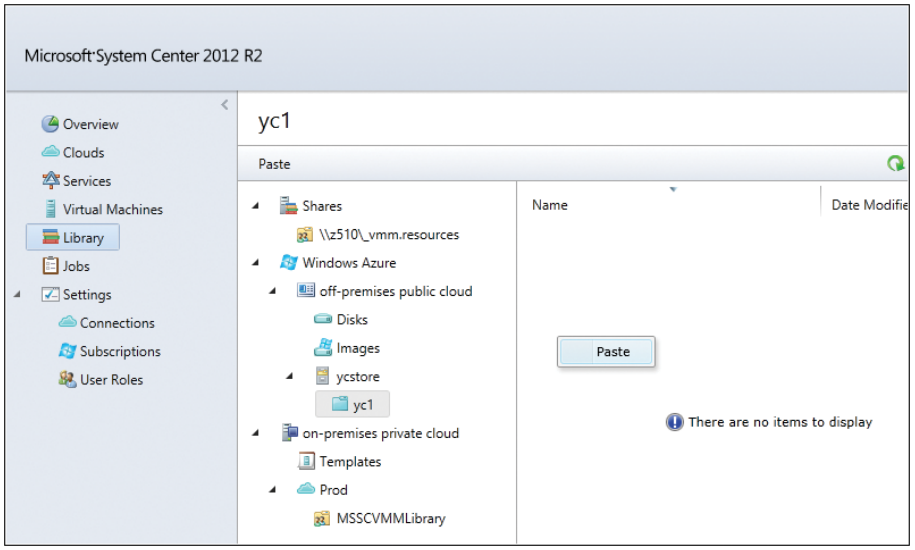


FIGURE 4-2 A VHD can be pasted into a Windows Azure storage container using App Controller.

When the copy/paste operation has completed, the VHD has been uploaded from the private cloud to the storage container in the Windows Azure subscription as shown in Figure 4-3.

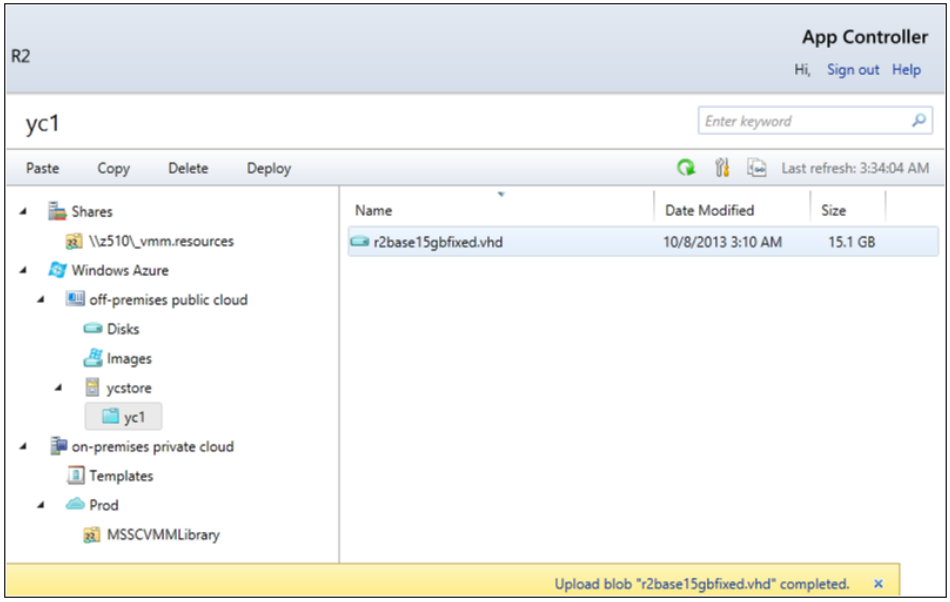


FIGURE 4-3 The VHD has been uploaded to the Windows Azure storage container using App Controller.

Selecting the Jobs workspace, as shown in Figure 4-4, confirms that the VHD file has been uploaded to blob storage in Windows Azure.

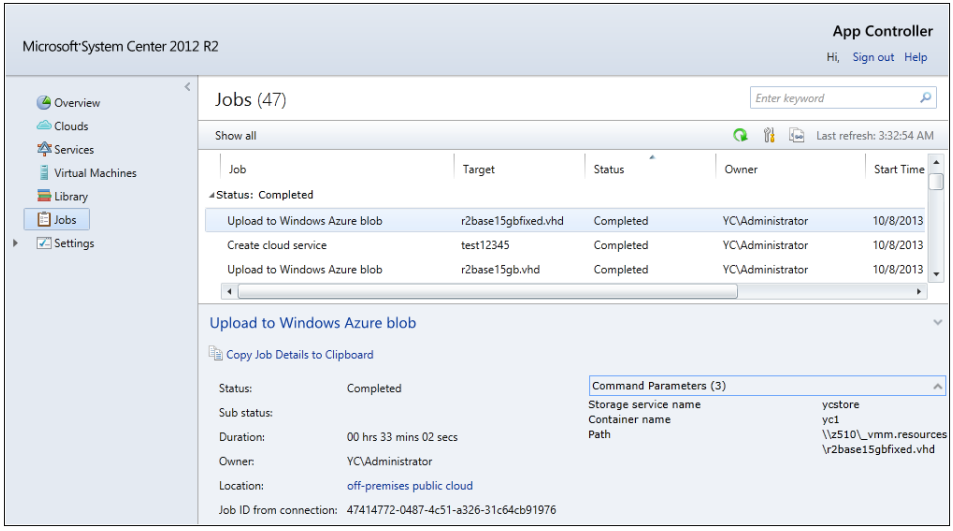


FIGURE 4-4 You can view the job history of uploading a VHD to a Windows Azure storage container using App Controller.

It's really just as simple as that, but there are a few details you need to understand concerning storage containers in Windows Azure. There are three types of Windows Azure storage: blob, table, and queue storage. To consume storage, you first need to create a storage account in Windows Azure. The storage account must be associated with a specific geographic region such as America, Europe, or Asia. At the time of this writing, a Windows Azure storage account can consume up to 100 TB of storage, and each Windows Azure subscription can create up to five storage accounts. VHD files must be stored in page blob format, and each page blob has a maximum size of 1 TB.

Another important thing you should realize is that although when you copy and paste a VHD into a Windows Azure storage container, the uploaded transaction appears to be committed to the target container relatively quickly; the actual uploading process takes place in the background and can take a long time. For example, in the home office setting where this walkthrough was performed, a 15 GB VHD took a little more than a half hour to upload to a storage container in Windows Azure. The next section demonstrates the process.

Deploying a cloud service in Windows Azure using an uploaded VHD

Once your sysprepped VHD has been uploaded to Windows Azure, deploying a new virtual machine in a Windows Azure cloud service from the uploaded VHD is easy using App Controller. You start the deployment process, as shown in Figure 4-5, by right-clicking the uploaded VHD file and selecting the Deploy option.

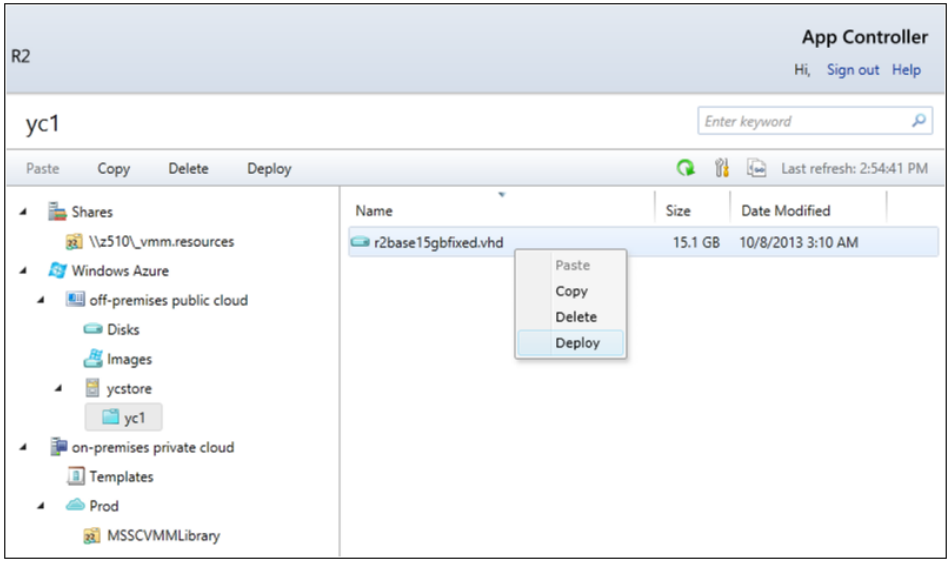


FIGURE 4-5 You can deploy an uploaded VHD in Windows Azure using App Controller.

A deployment wizard will then guide you through the process. While we've covered this previously in Chapter 3, we'll examine it here from a process point of view. Figure 4-6 shows the New Deployment Wizard with four configuration steps as follows:

1. **Public Cloud** This is used to specify the *destination* of the deployment process.
2. **Cloud Service** This is used to specify the *container* to which the payload will be deployed.
3. **Deployment** This is used to specify the *topology and logistics* for the deployment.
4. **Virtual Machine** This is used to specify the *payload* being deployed.

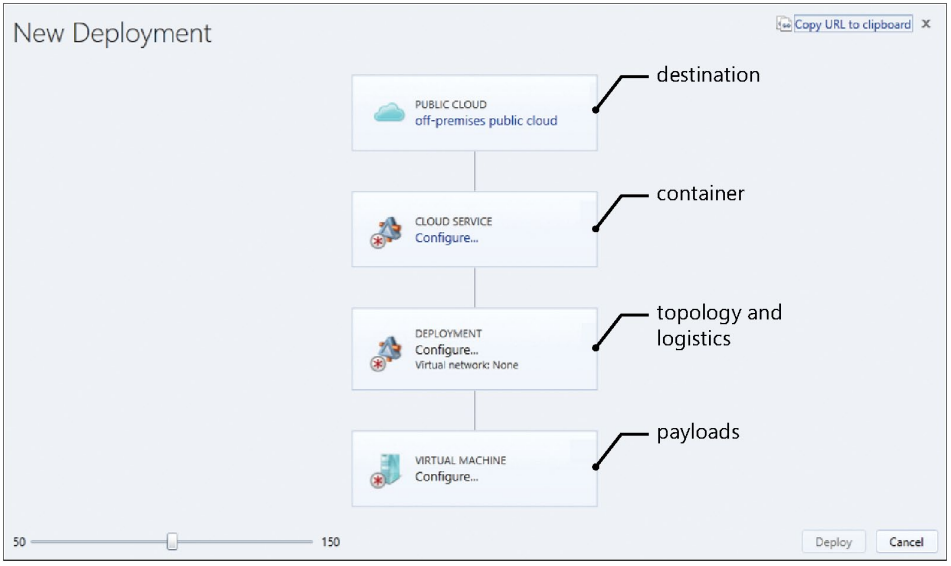


FIGURE 4-6 The New Deployment Wizard can be used for cloud service deployment in App Controller.

Let's now examine each of these steps in the deployment process in more detail.

Destination

Since you will be using Windows Azure for hosting the cloud service deployment, the destination will be your Windows Azure subscription. This means you need to select a VHD from a storage container in the currently connected Windows Azure subscription.

Container

Windows Azure cloud services act as containers (logical constructs) that associate all resources, references, and constraints concerning a delivery for a requested object. In the context of cloud computing, a service is a runtime concept that delivers objects which are capabilities. A service at an implementation level is a unit of delivery which is instantaneous and on demand.

In cloud computing environments like Windows Azure, services are the basic mechanism of delivery, not virtual machine instances, run-time environments, or applications (although each of these can be delivered like services as in IaaS, PaaS, and SaaS). What this means is that there must be a cloud service for every deployment instance. For example, with IaaS the objects that are delivered are virtual machines.

When performing a deployment, if there is not a service already implied, the wizard offers an opportunity to configure one. By clicking Configure under Cloud Service in Figure 4-6, a dialog box opens and allows you to select the cloud service you want to use for your deployment (see Figure 4-7).

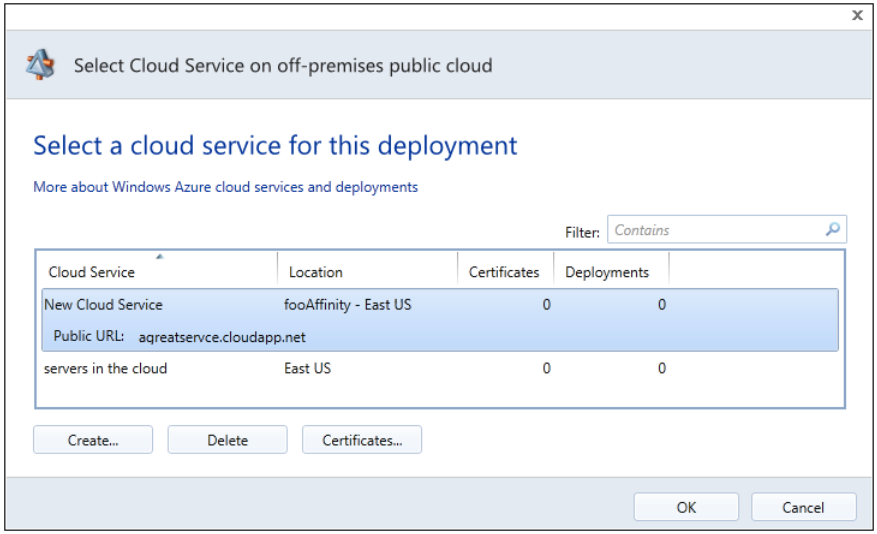


FIGURE 4-7 You can select a cloud service provider for the deployment.

MORE INFO **In the world of cloud computing, it's not just about managing virtual machines anymore; instead, it's about managing services. This concept is well explained in "A Memorandum to IT Leadership and Decision Makers" which is found at *http://aka.ms/memo*.**

An important consideration when planning the deployment of Windows Azure cloud services is the concept of an affinity group, which is an artifact used to facilitate deployments. An affinity group is a designation to a Microsoft data center in a specific region such as North America, Europe, or Asia. When you create a new virtual network it must be associated with an affinity group in order to ensure that all compute instances (virtual machines) associated with the virtual network are deployed in the same data center so they can be as close together as possible to optimize performance and help reduce transmission costs. Affinity groups can also be applied to storage accounts so that all the VHDs of virtual machines on

the same virtual network are kept near their associated compute instances. Cloud services can also be associated with affinity groups to ensure all resources deployed to the service are kept as close as possible in the data center or region designated by the group. A blog post explaining all this in more detail can be found at *http://aka.ms/AzureIaaSMethod*.

Topology and logistics

Once the destination and container have been specified, the next step is deciding on the network topology and logistics. For example, in the Windows Azure PaaS model you can chose to deploy the application (package) to either a staging or a production environment. A staging environment is a holding place for testing purposes and does not provide a virtual IP address, which means the deployed application is not exposed to the Internet. A production environment assigns a virtual IP address and therefore exposes the application to the Internet.

Windows Azure IaaS deployments, on the other hand, only provide a production environment, which explains why the Staging option button shown in Figure 4-8 is dimmed.

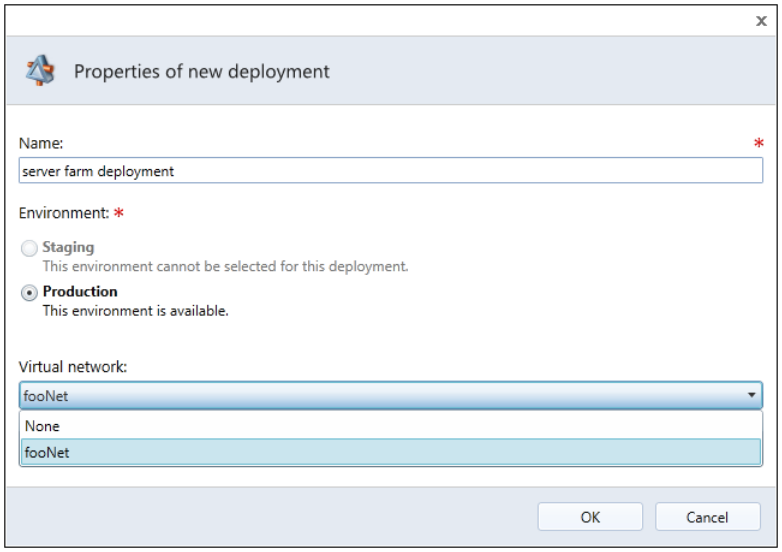


FIGURE 4-8 IaaS deployments are for production only.

Payload

The final step of the deployment process is defining the payload which, in this walkthrough, means specifying a virtual machine from the sysprepped VHD you uploaded from your private cloud to Windows Azure. Our intention is to deploy a sysprep VHD, which is basically a virtual machine image. Deploying the virtual machine involves configuring its General, Disk, and Network properties as described in this section.

General

The General page shown in Figure 4-9 includes an important property known as the availability set. You can either create a new availability set or select an existing one.

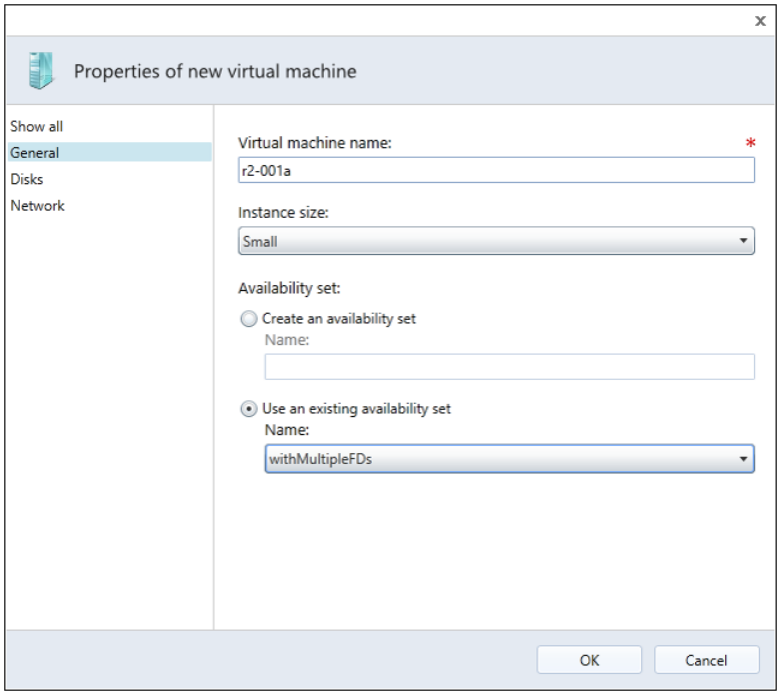


FIGURE 4-9 An example of the General page of a Windows Azure virtual machine.

Availability sets are designed to help eliminate single points of failure due to hardware problems. In data center environments like those used by Windows Azure, a server rack is considered a single point of failure (fault domain) because the top-of-the-rack switch is not redundant. If multiple virtual machine instances are deployed to a single server rack, they are in the same fault domain and are therefore not fault tolerant. When you configure virtual machines as an availability set, however, Windows Azure automatically eliminates single point of failure due to hardware by placing the virtual machine instances across multiple fault domains. (Windows Azure guarantees at least two fault domains for an availability set.) This concept is illustrated by the diagram in Figure 4-10.

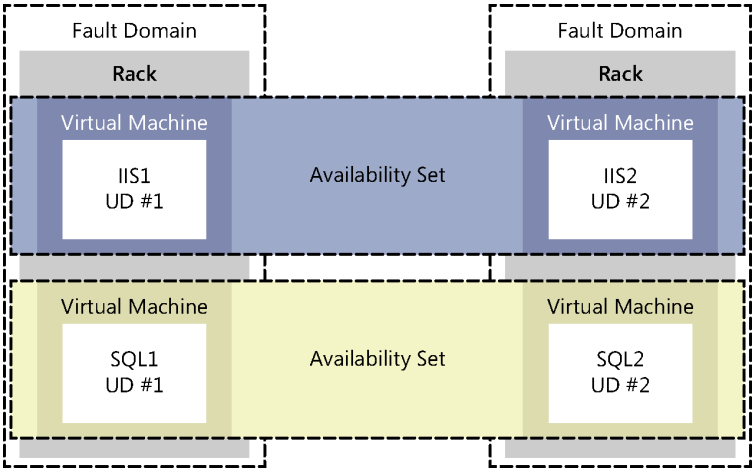


FIGURE 4-10 Windows Azure uses the concept of availability sets.

MORE INFO **For more information on fault domains and availability sets, see *http://aka.ms/wafdud*.**

Disks

By default, a Windows Azure virtual machine has two disks: C drive as the operating system disk and D drive for temporary storage and paging. When a virtual machine is relocated, the temporary disk is lost. Both the operating system and temporary disks are not intended for storing application data, however; that is what data disks are for.

A virtual machine in Windows Azure can have one or more data disks attached to it for storing application data. The deployment wizard can add an existing disk or create a new disk. For example, in Figure 4-11 you see datadisk1 being attached to the virtual machine being deployed.

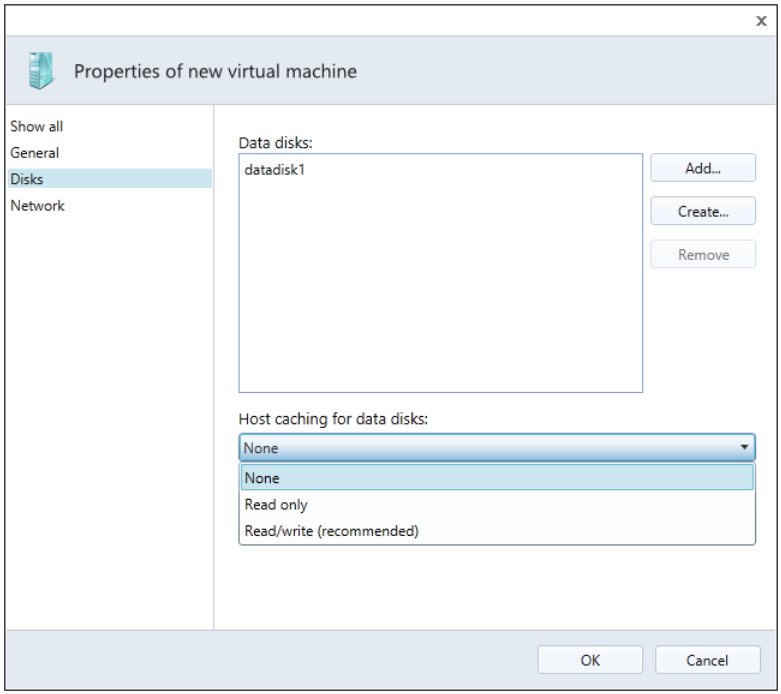


FIGURE 4-11 A view of the data disk settings of a Windows Azure virtual machine.

The number of data disks that can be attached to a virtual machine is determined by the size of the virtual machine and can range 1 data disk for a very small virtual machine to 16 disks for a very large one. Each disk is a page blob in Windows Azure Storage and can have a maximum size of 1 TB. As the size of a virtual machine increases, so does the number of data disks available, increasing the total data storage. And since the size of a virtual machine can be changed after the virtual machine has been deployed, so does the possible total data disk storage. Note that there are some additional considerations concerning the IP address assignment by Windows Azure when changing the size of a deployed virtual machine as detailed in the virtual network settings.

The Disks page also allows you to configure caching for data disks. By default, any data written to or from the operating system disk is first cached locally followed by reading from and writing back to the associated blob storage. This is because the data read from

and written to the operating system is relatively small and can possibly fit in the cache for performance gain. Caching is not enabled by default on data disks. That's because applications, for instance an SQL database, tend to process a much larger amount of data than the disk cache can hold. But for certain types of applications, you can enable caching for data disks as shown in Figure 4-11.

Network settings

If the virtual machine is being deployed to a virtual network, the configured subnets can be selected from the drop-down list on the Network page as shown in Figure 4-12. The Remote Desktop Protocol (RDP) endpoint is set by default to the private port 3389 and a blank (that is, random) public port. Additional endpoints can also be added and the figure shows one intended for an SQL connection on private port 1433 and public port 9296.

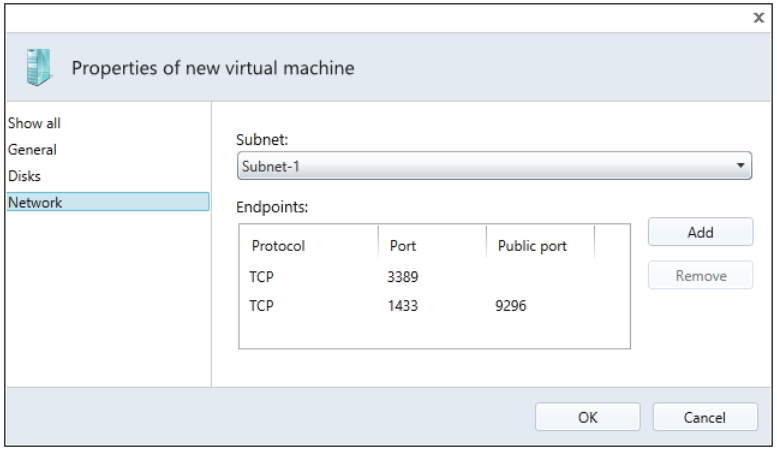


FIGURE 4-12 A view of the network settings for a Windows Azure virtual machine.

Completing the deployment

Once the deployment has been configured as described in the example above, the Deployment Wizard looks like Figure 4-13. The resulting deployment will create a new cloud service that has one virtual machine attached to the specified virtual network and created from the previously uploaded VHD file.

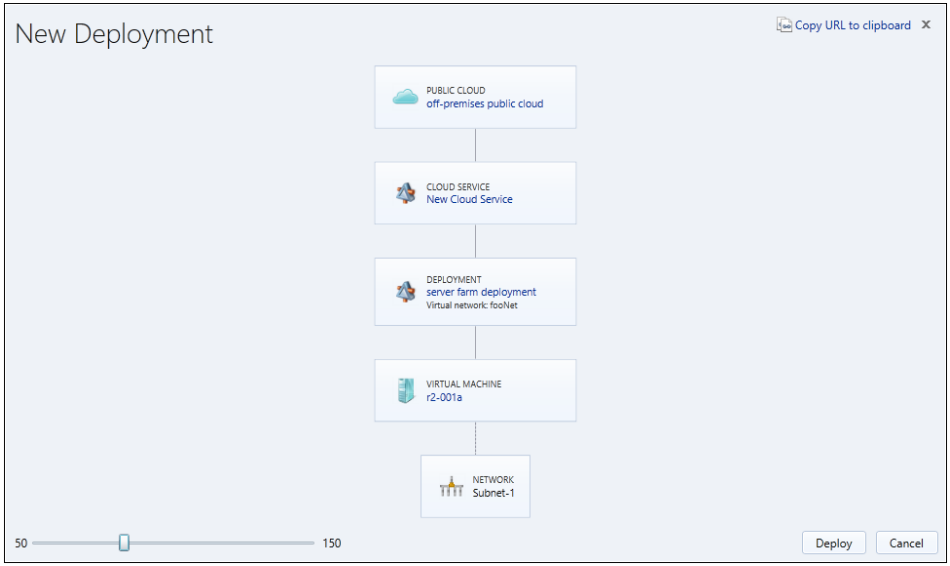


FIGURE 4-13 Sample settings for a Windows Azure virtual machine deployment based on an uploaded VHD.

Upon completion of the deployment, the job history displays the information shown in Figure 4-14.

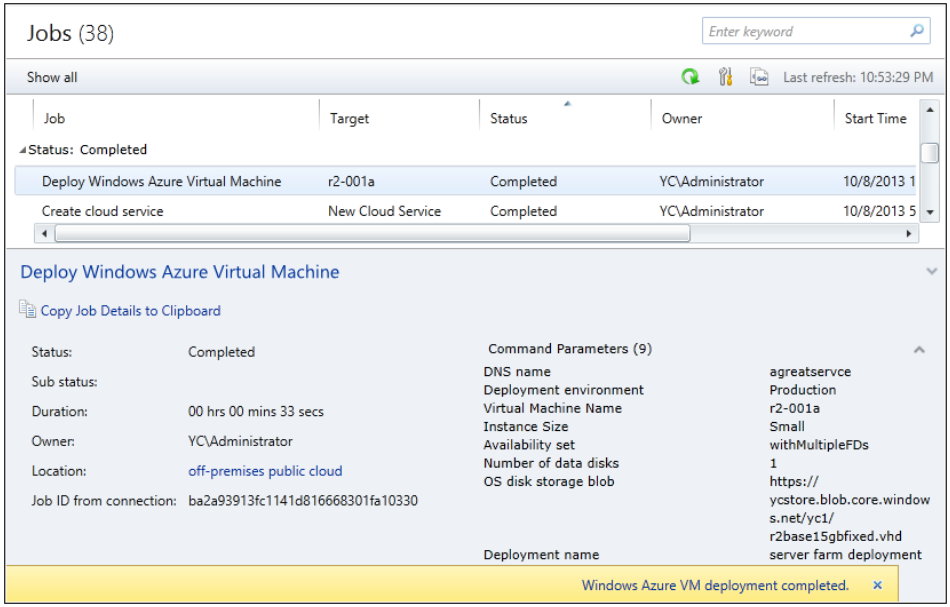


FIGURE 4-14 The job details of a virtual machine deployment based on an uploaded VHD.

The deployed virtual machine is then listed in the Virtual Machines workspace with the connection name Windows Azure indicating where the virtual machine has been deployed. If you right-click the virtual machine, a menu option displays that allows you to use Remote Desktop to connect to the guest operating system of the virtual machine from the App Controller console (see Figure 4-15).

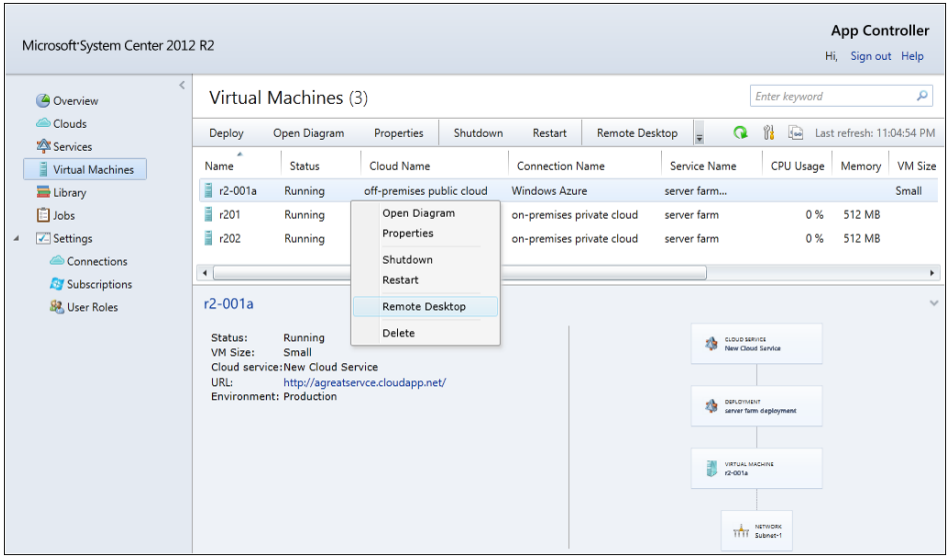


FIGURE 4-15 A Windows Azure virtual machine has a Remote Desktop option in App Controller.

Copying virtual machines from VMM to Windows Azure

App Controller also simplifies the task of copying entire virtual machines between your VMM-based private cloud and the Windows Azure public cloud or a third-party hosted cloud environment. This allows Windows Azure and hosted clouds to function as an off-premises extension of your on-premises data center, thus enabling new scenarios and exciting opportunities for enterprise computing. Figure 4-16 illustrates the basic concepts involved. In the figure, a virtual machine running on-premises, and which is stored and managed by VMM, is to be copied to an off-premises cloud environment, which in this case is your Windows Azure subscription. A similar scenario would be to copy a virtual machine running in Windows Azure back down to your VMM-based private cloud.

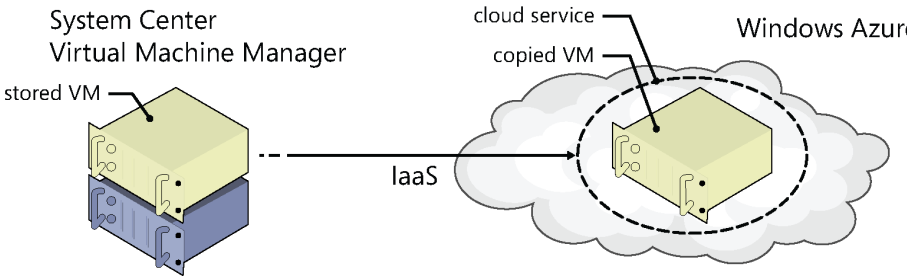


FIGURE 4-16 A conceptual model of copying a virtual machine between a private cloud and Windows Azure.

Before you can copy a virtual machine from a VMM-based private cloud to Windows Azure, you must first establish a secure connection between App Controller and the associated VMM server that manages the private cloud where the virtual machine resides. You must also establish a secure connection between App Controller and the Windows Azure cloud service to which you want to copy your virtual machine. You can review Chapters 2 and 3 as needed for the steps to establish such connections.

Once secure connections have been established, the virtual machine must first be stored before it can be copied. This action places the virtual machine in a saved state and then exports the virtual machine to the location where the stored virtual machine path points, which is in the Library workspace. To store the virtual machine, you simply right-click it and select Store as shown in Figure 4-17.

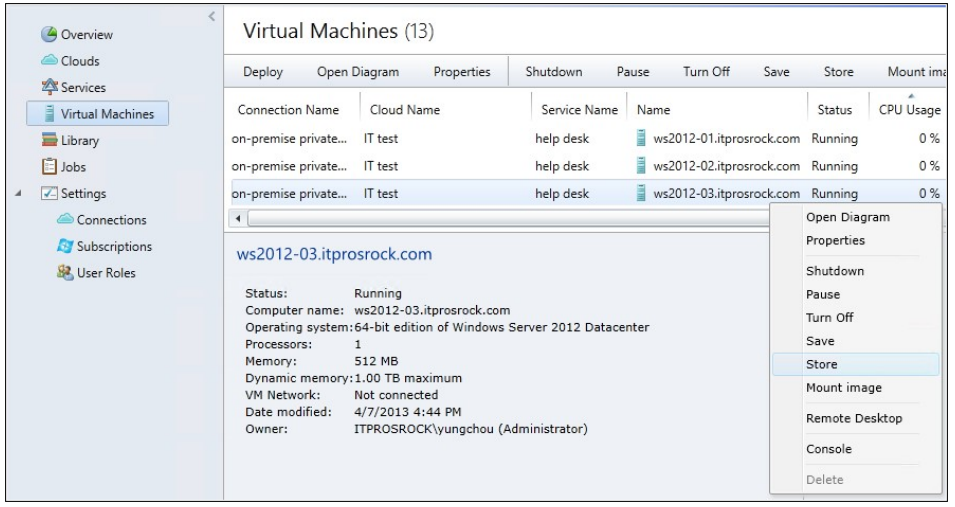


FIGURE 4-17 You must first store a virtual machine in App Controller before copying it.

The Store process saves the current state of the virtual machine and exports the virtual machine to the location which the stored virtual machines path points to, as shown in Figure 4-18. This gets the virtual machine ready for the Copy operation.

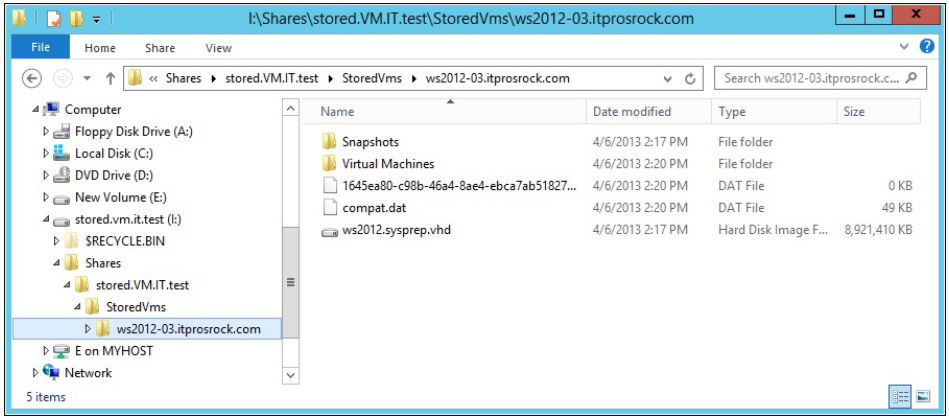


FIGURE 4-18 An example of the stored virtual machine path.

The Copy operation is then initiated by right-clicking the stored virtual machine and selecting Copy as shown in Figure 4-19.

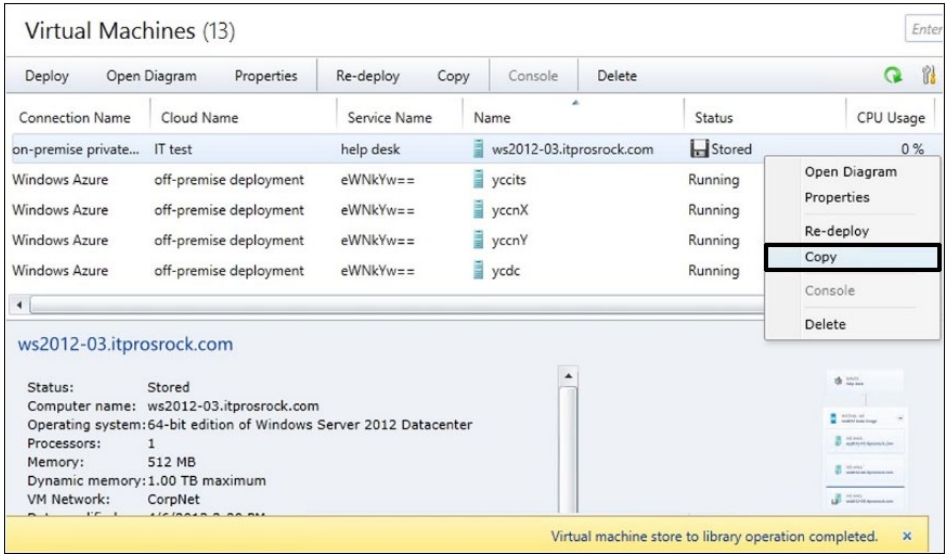


FIGURE 4-19 An example of copying a stored virtual machine in App Controller.

What is actually happening under the hood when the Copy operation is performed is a bit complex, as illustrated by the diagram in Figure 4-20. Specifically, a designated cloud service in Windows Azure is first either identified or created on demand. An associated Windows storage account is then either identified or created to provide storage space for the virtual machine files. Then, upon finishing configuring the cloud service, the virtual machine files are copied to the storage account after which the virtual machine is deployed to the cloud service. The virtual machine is then brought to a running state.

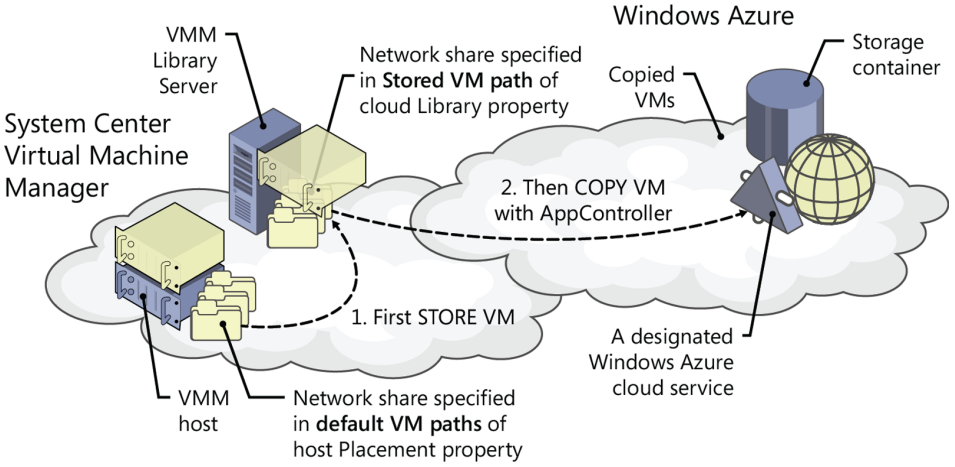

Windows Azure
VMM Library Server
Network share specified in **Stored VM path** of cloud Library property
Copied VMs
Storage container
System Center Virtual Machine Manager
2. Then COPY VM with AppController
1. First STORE VM
A designated Windows Azure cloud service
VMM host
Network share specified in **default VM paths** of host Placement property

FIGURE 4-20 It is important to understand the process of copying a virtual machine from VMM to Windows Azure.

The steps performed during the Copy process can be viewed in the job history as shown in Figure 4-21.

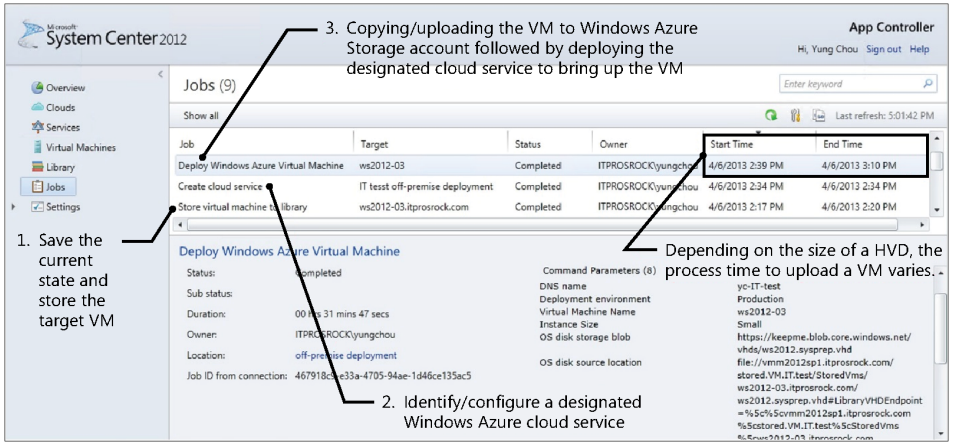

System Center 2012
App Controller
3. Copying/uploading the VM to Windows Azure Storage account followed by deploying the designated cloud service to bring up the VM
Jobs (9)
Deploy Windows Azure Virtual Machine
Create cloud service
Store virtual machine to library
1. Save the current state and store the target VM
Depending on the size of a HVD, the process time to upload a VM varies.
2. Identify/configure a designated Windows Azure cloud service

FIGURE 4-21 The copy virtual machine process has a viewable job history.

Once the Copy operation has completed, the running virtual machine can then be managed using the Windows Azure Management Portal as shown in Figure 4-22.

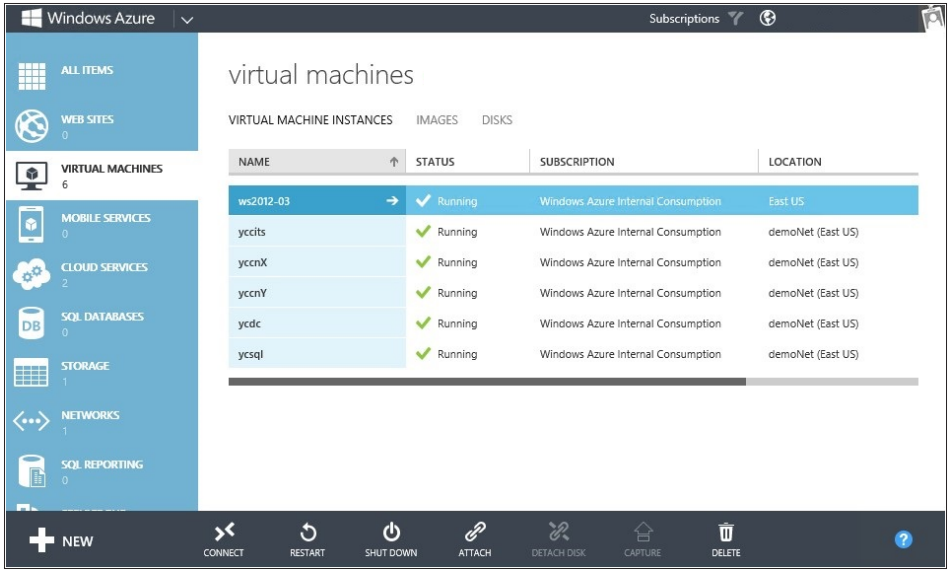


FIGURE 4-22 You can manage the copied virtual machines in Windows Azure.

An administrator who is logged on to App Controller and has suitable privileges will also see the same virtual machine listed as a resource of the associated Windows Azure subscription. In other words, you can also access and manage the copied virtual machine using App Controller; you don't need to use the Windows Azure Management Portal to manage the virtual machine if you don't want to.

CHAPTER 5

App Controller cmdlets

This chapter provides an introduction to the Windows PowerShell cmdlets for Microsoft System Center 2012 App Controller. Some of these cmdlets focus on connectivity and related issues with target hosts while others facilitate interoperability with System Center 2012 Virtual Machine Manager (VMM) libraries and the Windows Azure public cloud. App Controller cmdlets can also make it easier to move workloads between different clouds. The topics covered in this chapter include:

- How App Controller cmdlets work
- Importing the App Controller module
- Connecting with the App Controller server, VMM, and Windows Azure
- Adding a library share to copy and paste resources between clouds
- Adding a VHD to a Windows Azure storage account container
- Adding a VHD to a Windows Azure image store
- Acquiring a VHD from a virtual machine, template, or the VMM library

How App Controller cmdlets work

When adding a connection for a Windows Azure subscription, there are a number of required fields. As Figure 5-1 shows, both the subscription ID and a local management certificate in PFX format are required to establish a secure connection. As demonstrated earlier in Chapter 3, "Managing public clouds," the local management certificate is paired up with an already uploaded x.509 certificate for the intended Windows Azure subscription so that communications between App Controller and the Windows Azure subscription can be carried out in a private and secure fashion. App Controller cmdlets also operate via an established connection with a Windows Azure subscription, and this is why a requirement for running App Controller cmdlets is that you first set the context by connecting to a target App Controller server for employing a previously established connection with the Windows Azure subscription.

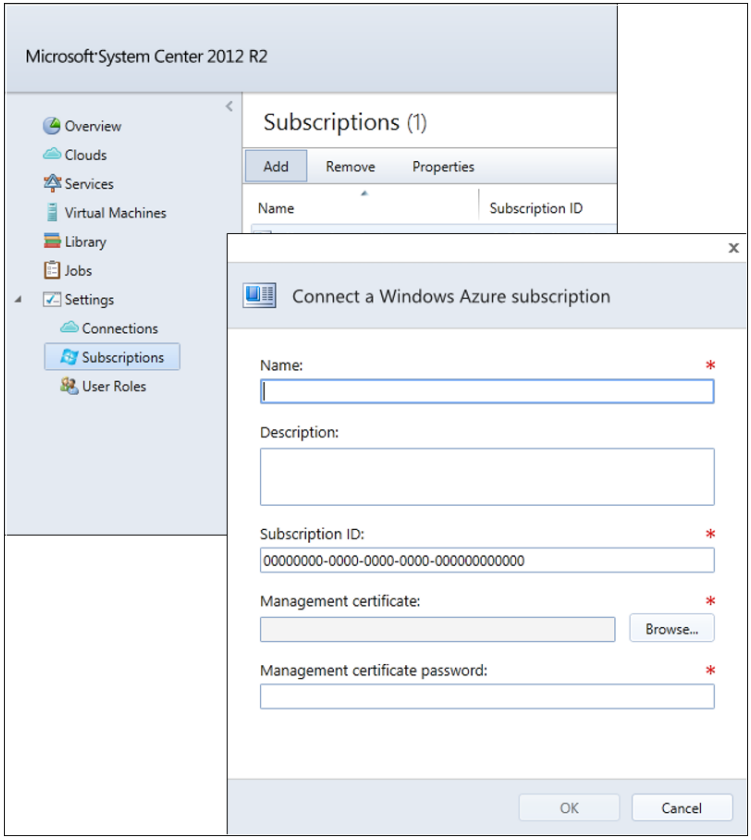


FIGURE 5-1 The Connect A Windows Azure Subscription dialog box displays five fields.

Why App Controller cmdlets?

With Windows PowerShell, you can connect to an instance of App Controller and take advantage of an already established connection with a Windows Azure subscription in order to manage resources without the need of having to use the App Controller console. For frequently performed tasks and long transactions, such as uploading large VHDs, streamlining and automating the tasks using App Controller PowerShell cmdlets is easy to implement and helps increase efficiency and repeatability.

Importing the AppController module

A quick way to access App Controller cmdlets is to click the application tile named "Windows PowerShell module for App Controller" which is created on the Start screen when App Controller is installed. Alternatively, within any Windows PowerShell session you can simply import the App Controller PowerShell module by running the following command:

```
Import-Module -Name AppController
```

There are also other Windows PowerShell modules, such as virtualmachinemanager and Hyper-V, that App Controller administrators might find useful. You can use Get-Module with -ListAvailable to find out which modules have been imported and which are installed and available for your current Windows PowerShell session. You can also use Update-Help as needed to ensure that the list of available modules is up to date.

As a side note, Get-Module and Import-Module are now the main tools for managing an IT environment that includes various platforms and hybrid deployment solutions. Many IT professional use these cmdlets to dynamically import modules needed for particular operations. The following sample query lists available modules in tabular form. Note that this command may run for a few seconds before it finally presents the results:

```
Get-Module -ListAvailable | Out-GridView
```

There are 29 cmdlets included in the AppController PowerShell module as revealed by the following Windows PowerShell command:

```
Get-Command -Module AppController
```

Figure 5-2 shows the results of running this command. Note all App Controller cmdlets have the prefix SCAC in the noun part. There are a number of different Get operations available which are typically used to acquire information from a connected host before App Controller can properly operate on objects, and this will be demonstrated in some of the examples later in this chapter.

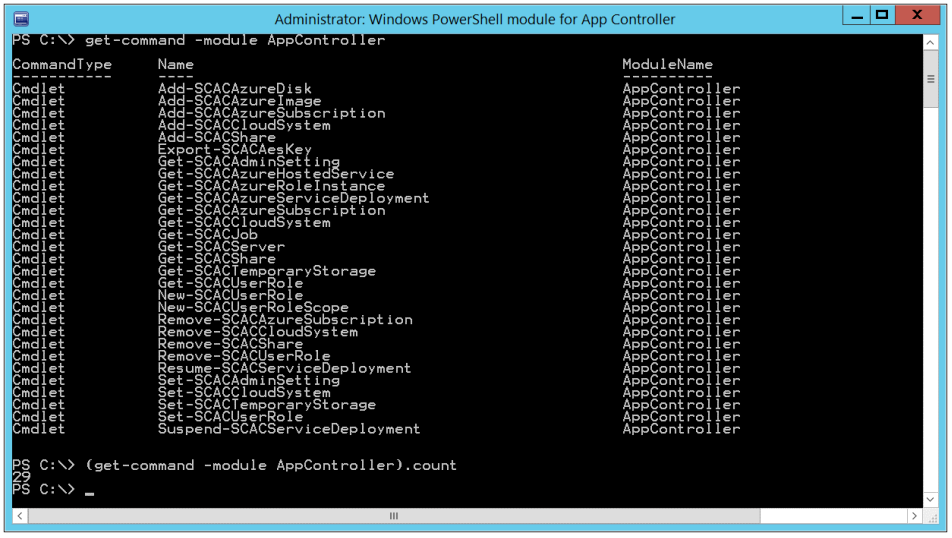


FIGURE 5-2 A view of the cmdlets included in the AppController module.

Connecting with the App Controller server

Prior to running any other App Controller cmdlets, the current Windows PowerShell session much first establish a connection with a target App Controller server. The process starts with getting the user credentials with which to establish the connection with the server. This is in essence identical to connecting interactively, as in such a case the user will enter his credentials from the log-on screen when accessing App Controller.

In this example, you will use the Get-SCACServer cmdlet to establish a connection with the specified App Controller server:

```
$Cred = Get-Credential
$SCAC = 'myTargetVMMServerURL'
Get-SCACServer -ServerName $SCAC -Credential $Cred
```

Get-Credential will open a dialog box prompting for the user name and password (see Figure 5-3). Once these have been entered, the cmdlet creates a credential object that represents the user's credentials. The variable, $SCAC here, is the App Controller's URL. In this example, you have App Controller installed on a VMM server named vmm2012r2.hc.lab.

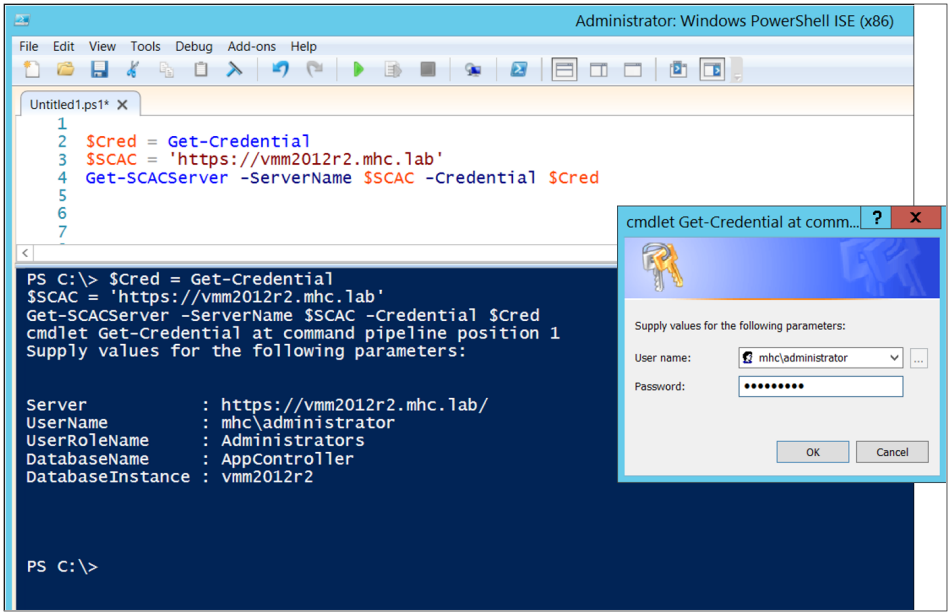


FIGURE 5-3 Enter your credentials to connect to App Controller.

Once a connection to the App Controller server has been established, you can now start bringing in VMM, Windows Azure, and third-party hosts.

Connecting to VMM

App Controller is part of the System Center family and is tightly integrated with VMM. Connecting to VMM is essential for enabling much of the functionality in App Controller since VMM provides the clouds, virtual machines, services, library servers, shares, role-based security, and so on.

The Add-SCACCloudsystem cmdlet includes a -VMM switch for connecting to a VMM server. In this example, the VMM server is again vmm2012r2.mhc.lab:

```
Add-SCACCloudsystem `
    -Name 'on-premises private cloud' `
    -VMMServerName 'vmm2012r2.mhc.lab' `
    -Port 8100
```

> ***NOTE*** **For connecting with a third-party cloud system, omit the -VMM switch.**

When you add a VMM connection, the resources that can be managed by the authorized user are based on their user role profile as defined in the Settings workspace of the VMM console as shown in Figure 5-4.

Should removing a VMM server or other cloud system later become necessary, you should first get the object by specifying the name of the cloud system before carrying out the remove operation as shown here:

```
$CloudSys = Get-SCACCloudSystem -Name 'on-premises private cloud'
Remove-SCACCloudSystem -CloudSystem $CloudSys
```

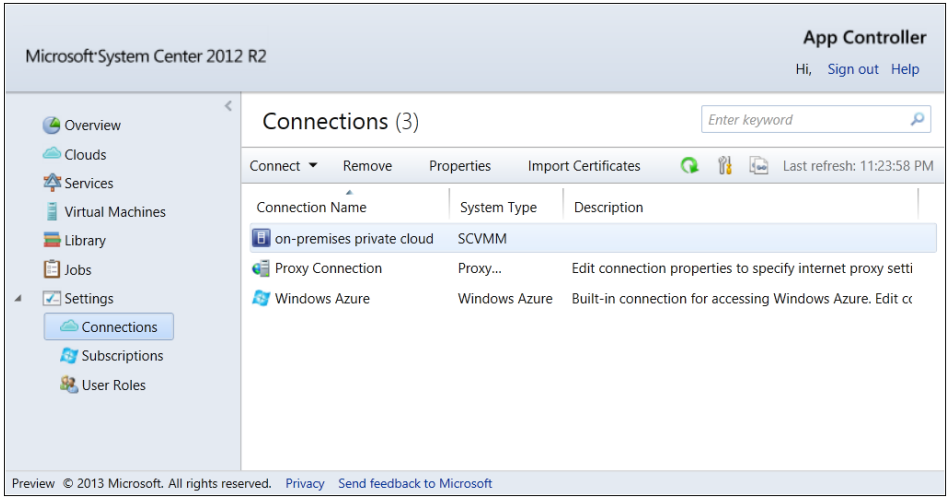


FIGURE 5-4 App Controller allows you to manage a VMM connection.

Connecting to Windows Azure

One of the most exciting capabilities of App Controller is its ability to connect not only with on-premises VMM-based private clouds but also with off-premises cloud facilities such as the Windows Azure public cloud or a cloud system at a third-party hosting company. While much of the IT industry is now transitioning to cloud computing, this is not a scratch-and-replace process. Enterprise IT is and will remain primarily a mixed deployment model that involves solutions from multiple vendors. The emerging IT business model is therefore the hybrid cloud deployment scenario. Being able to manage resources deployed to both private and public clouds, both on premises and off, enables new scenarios that otherwise might be financially cost-prohibitive or technically unfeasible.

You can use App Controller cmdlets to connect to Windows Azure provided there is a certificate pair as follows:

- An x.509 certificate uploaded to the certificate store in the Windows Azure Management Portal / Settings workspace for the intended Widows Azure subscription.
- A corresponding PFX format certificate that is password-protected and made locally available for initiating a secure connection with the Windows Azure subscription.

These paired certificates can be self-signed, based on an internal PKI, or purchased from a third-party certificate authority (CA).

To connect to Windows Azure with App Controller cmdlets, one must first acquire an x.509 certificate. Because installing App Controller also installs Internet Information Services (IIS), it's easy to use the IIS Manager Server Certificates page to either generate a self-signed certificate for testing or to create a certificate request file for issuing a production-ready one from your enterprise PKI. Figure 5-5 shows the Server Certificates page in the IIS Manager, which has links to generate a self-signed certificate and create a certificate request. The process is described in detail in Chapter 3.

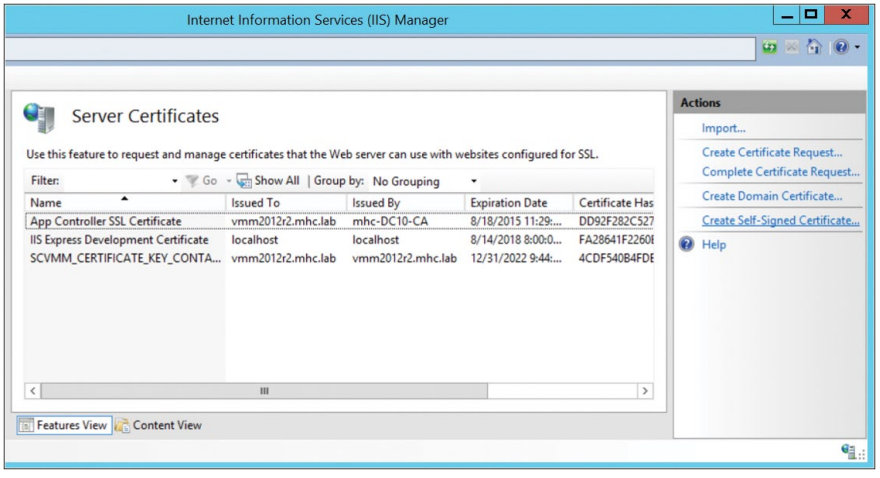


FIGURE 5-5 The Server Certificates page has links to create certificate requests and self-signed certificates.

Once you have acquired an x.509 certificate for connecting with Windows Azure, right-click the certificate to install it in the local certificate store on the server. Next, use the Certificates snap-in to export the certificate to PFX format with the private key as shown in Figure 5-6. Then log on to the target Windows Azure subscription and upload the x.509 certificate to the certificate store by selecting the Settings workspace of the Windows Azure Management Portal.

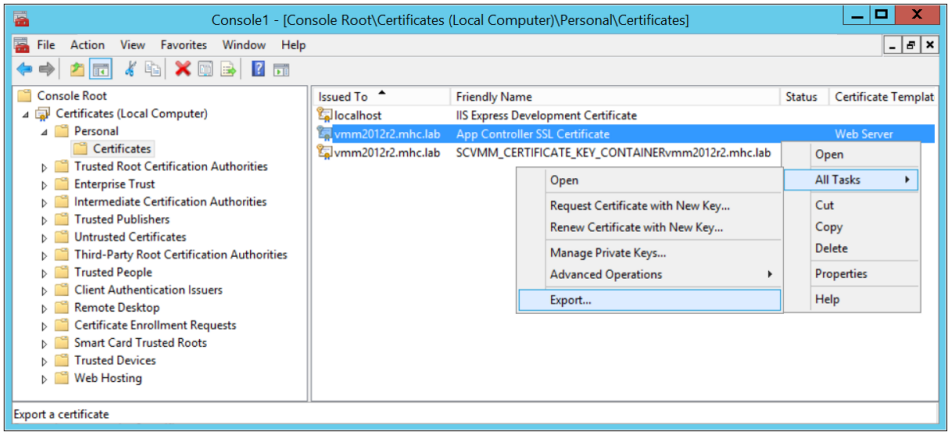


FIGURE 5-6 The Certificates snap-in allows you to export a certificate.

The cmdlets to connect App Controller with a Windows Azure subscription can be condensed into the following two statements:

```
$myPFXPwd = (ConvertTo-SecureString 'password of the PFX file' -AsPlainText -Force)
Add-SCACAzureSubscription -Name 'off-premises deployment' `
    -Id 'this is the Windows Azure subscription id' `
    -ManagementCertificatePath 'C:\_cert\vmm2012r2.mhc.lab.pfx' `
    -ManagementCertificatePassword $myPFXPwd `
    -Description 'IaaS, PaaS, and SaaS'
```

The first statement converts the password for accessing the PFX certificate from plain text to a secure string. The second statement then uses the secure string for accessing the PFX format certificate. With the subscription ID, the PFX certificate, and the password specified, a secure connection with the Windows Azure subscription can be established. Note that the paired x.509 certificate shown in Figure 5-7 needs to be uploaded to the Settings workspace beforehand.

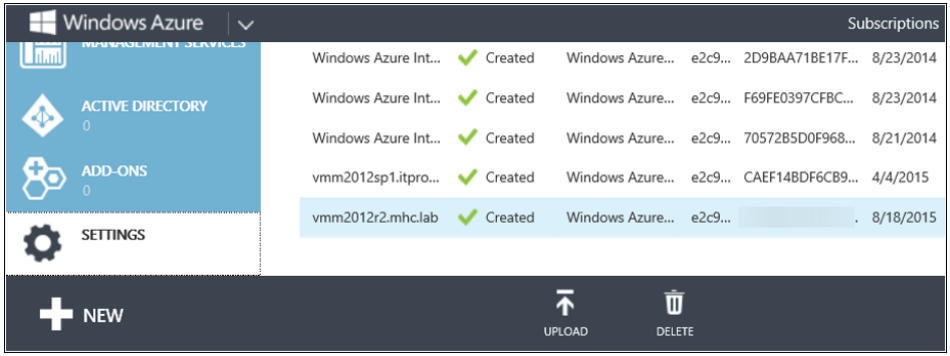


FIGURE 5-7 The Settings page lists the Windows Azure management certificates for secure connections.

If you now log on to the App Controller console, the Subscriptions page will display the configured Windows Azure connection and the associated services, virtual machines, and other resources will be presented in corresponding groups. Figure 5-8 lists the VHDs for a Windows Azure subscription and shows that you can operate from App Controller directly on a VHD deployed in Windows Azure.

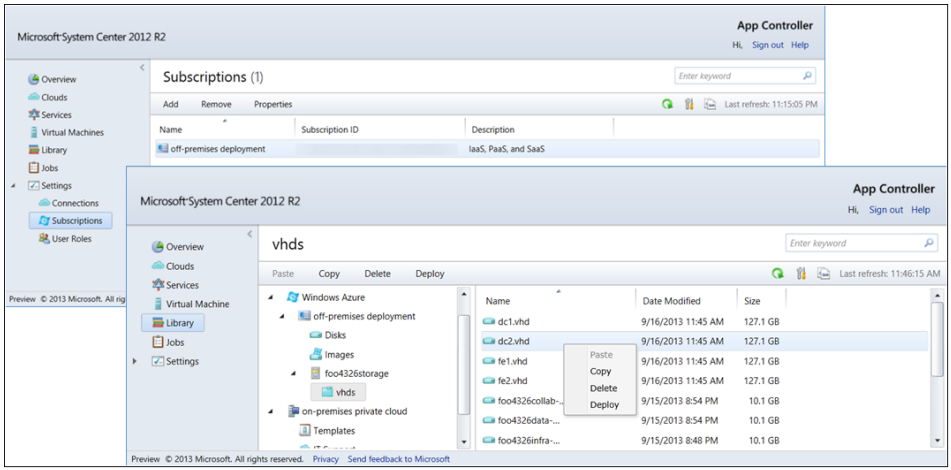


FIGURE 5-8 An example of operating from App Controller on resources deployed to Windows Azure.

Upon establishing a connection with Windows Azure, the authorized user can then manage the services, virtual machines, and storage associated with this subscription using the remainder of the App Controller cmdlets listed previously in Figure 5-2.

Removing a connection with Windows Azure is also a straightforward process. You simply retrieve and then remove the subscription object as shown here:

```
$Subscription = Get-SCACAzureSubscription -Name 'off-premises deployment'
Remove-SCACAzureSubscription -Subscription $Subscription
```

Adding a library share to copy and paste resources between clouds

App Controller also enables you to create library shares using the console or cmdlets. These shares are not the same, however, as those configured for library servers managed by VMM. The shares that VMM manages become visible based on user role profiles via a VMM connection in App Controller. However, App Controller library shares are user-defined, ad hoc, and easy to create as shown below:

```
Add-SCACShare -Path '\\dc10\AC-Share'
Add-SCACShare -Path '\\vmm2012r2\btv'
```

If you run the two previous commands, the shares will appear in the Library workspace in the App Controller console.

A library share is basically a tool for performing certain complicated tasks in a simple and straightforward manner. For instance, to move a VHD from an on-premises location to your Windows Azure cloud is often tedious work. But by using App Controller this is now as simple as performing a copy-and-paste operation with a library share. You simply place the target VHD on a network share and then expose the share as a library share in App Controller. Then, as shown in Figure 5-9, you copy the VHD from the library share and paste it into a storage container of your Windows Azure subscription.

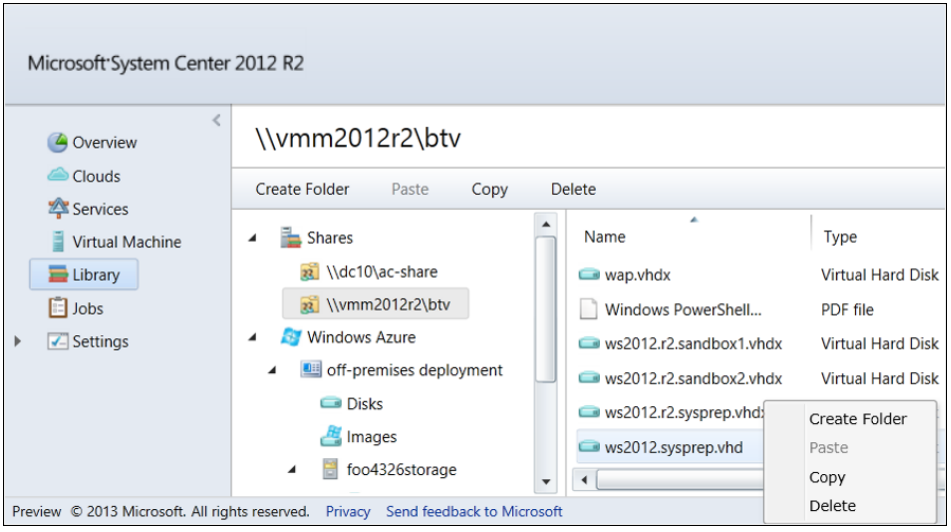


FIGURE 5-9 Self-service and ad hoc operations on resources across premises are performed by an App Controller user.

The behind-the-scenes process of securely transferring gigabytes of data across geo-regions is certainly much more complex than it appears. But with App Controller this has become a simple mouse-click process that can be carried out in a self-service fashion on demand. And

this is not just for relocating workloads, but for using the cloud as an extension of what you do every day, from on-premises operations to business continuity and disaster recovery.

You can also remove library shares easily when their work is done:

```
$Share = Get-SCACShare | where {$_.SharePath -eq '\\vmm2012r2\btv'}
Remove-SCACShare $Share
```

Note that like other operations, you first query to get the specific library share object and then you remove it.

Adding a VHD to a Windows Azure storage account container

The ability to connect with and operate on resources deployed in VMM-based private clouds and Windows Azure public clouds at the same time is a clear advantage of App Controller and opens up many exciting scenarios. Using clouds to extend on-premises deployments can be strategic for businesses and is an emerging IT computing model. As more and more workloads are now running in virtual machines, uploading VHDs to the cloud for backups, restores, production, testing, development, and training are becoming a daily reality for IT, and the App Controller cmdlets make it easy for you to automate this process.

For example, let's see how to add a VHD to a Windows Azure storage account container using Windows PowerShell. Begin by obtaining the target Azure subscription object (that is, the Windows Azure connection object configured in App Controller) using the Get-SCACAzureSubscription cmdlet. You also need to decide where to store the VHD and with what name once you have uploaded it to your Windows Azure subscription. Windows Azure storage follows a naming standard like this:

https://theStorageAccountName.blob.core.windows.net/theContainerName/vhdName

The actual process of uploading the VHD to the storage account container is carried out by the Add-SCACAzureDisk cmdlet. The following App Controller statements provide an example of how to accomplish this task:

```
Import-Module virtualmachinemanager
$Subscription = Get-SCACAzureSubscription -Name 'off-premises deployment'
$thisBlob =
    'https://foo4326storage.blob.core.windows.net/vhds/yungchou.ws2012.sysprep.vhd'
Add-SCACAzureDisk -Name 'WS2012' `
    -DisplayName 'Yung Chou''s Windows Server 2012 Image' `
    -Cloud $Subscription `
    -StorageBlob $thisBlob `
    -SourcePath '\\vmm2012r2\btv\ws2012.sysprep.vhd' `
    -OperatingSystem 'Windows' `
    -Force
```

Notice that the Add-SCACAzureDisk cmdlet first requires importing the VMM module (that is, virtualmachinemanager) into a current Windows PowerShell session in order to run this cmdlet.

Figure 5-10 shows the result after completing execution of the previous Windows PowerShell statements. The target VHD has been uploaded to Windows Azure, and in the Windows Azure Management Portal the Storage workspace displays the VHD with the correct URL. Meanwhile in App Controller, the VHD correctly displays in the vhds container under the storage account, foo4326storage.

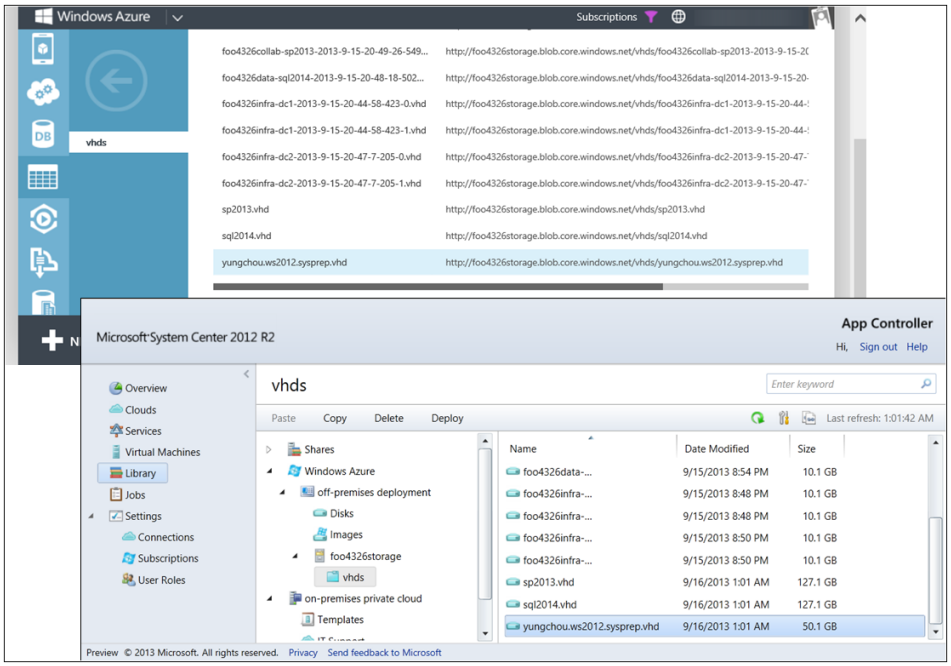


FIGURE 5-10 An example of uploading a VHD to Windows Azure with App Controller.

Adding a VHD to a Windows Azure image store

As a second example, let's use Windows PowerShell to upload the image myImage.vhd to blob storage in the target Windows Azure subscription and rename the image as ws2012.image.vhd:

```
Import-Module virtualmachinemanager
$Subscription = Get-SCACAzureSubscription
    -Name 'myAppControllerAzureConnectionDisplayName'
Add-SCACAzureImage `
    -Cloud $Subscription `
    -DisplayName 'myImage' `
```

```
    -Name 'myImage' `
    -SourcePath 'myImage.vhd' `
    -StorageBlob
        'http://myStorageAccountName.blob.core.windows.net/vhds/ws2012.image.vhd' `
    -OperatingSystem 'Windows' `
    -Force
```

In the examples in this section and the previous one, the source VHD was readily available. In reality, however, the target VHD may not be directly available. So the question arises: How can you acquire a target VHD object? The next section shows you how.

Acquiring a VHD from a virtual machine, template, or the VMM library

An authorized user can use the Get-SCVirtualHardDisk cmdlet to get a VHD object from a virtual machine, a template, or a VMM library. The VHD can be a Windows-based .vhd file, a Citrix XenServer-based .vhd file, or a VMware-based.vmdk file. The VHD can be stored as a standalone object in the VMM library server, included in a template, or deployed or stored with a virtual machine.

To get a named VHD object from a VMM library server, you could use the following statement:

```
$VHD = Get-SCVirtualHardDisk -VMMServer 'myVMMserverFQDN' |
    where { ($_.Name -eq 'myNamed.vhd') `
    -and ($_.LibraryServer.Name -eq 'theLibraryServerWhereTheVHDis') }
```

In this example, all of the VHD objects are first retrieved from the VMM server. Then the objects are filtered based on both the name and the associated library server. Figure 5-11 shows how to acquire the VHD object web.server.base.image.vhd kept in the VMM library server, vmm2012r2.mhc.lab.

To acquire the VHD object from a specific template of a VMM server, you need to first discover all the templates on the server. Then you filter them based on the name of the target template before retrieving the associated VHD object as shown in the following statement:

```
$VHD = Get-SCVMTemplate -VMMServer 'myVMMserverFQDN' |
    where {$_.Name -eq 'myTargetTemplateName'} | Get-SCVirtualHardDisk
```

Figure 5-12 illustrates using the previous statement to acquire the VHD object employed by the template “ws2012 base image” on the VMM server vmm2012r2.mhc.lab.

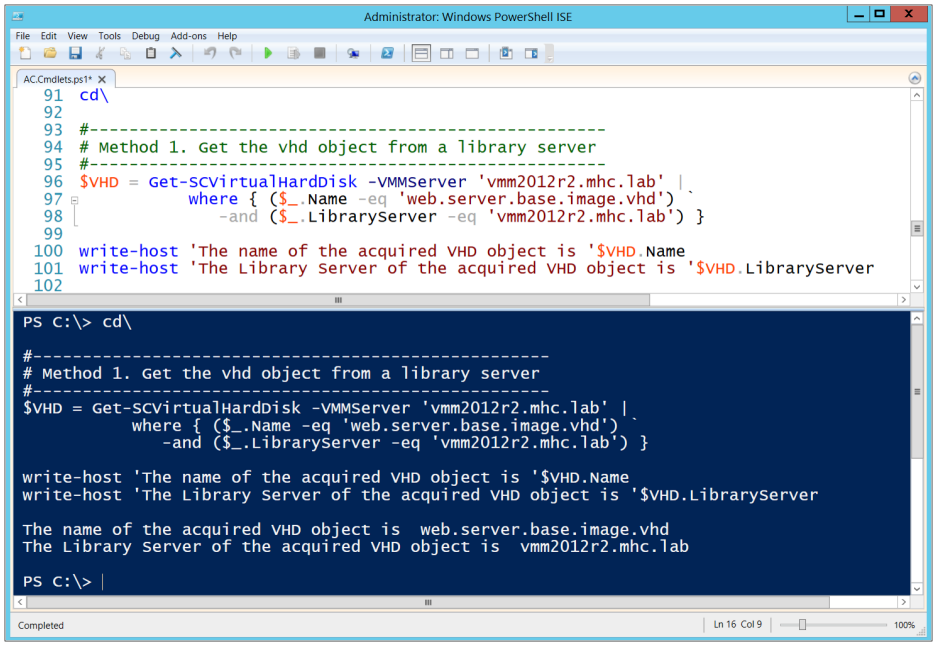

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
AC.Cmdlets.ps1*
91 cd\
92
93 #---
94 # Method 1. Get the vhd object from a library server
95 #---
96 $VHD = Get-SCVirtualHardDisk -VMMServer 'vmm2012r2.mhc.lab' |
97 where { ($_.Name -eq 'web.server.base.image.vhd') `
98 -and ($_.LibraryServer -eq 'vmm2012r2.mhc.lab') }
99
100 write-host 'The name of the acquired VHD object is '$VHD.Name
101 write-host 'The Library Server of the acquired VHD object is '$VHD.LibraryServer
102
PS C:\> cd\
#---
Method 1. Get the vhd object from a library server
#---
$VHD = Get-SCVirtualHardDisk -VMMServer 'vmm2012r2.mhc.lab' |
where { ($_.Name -eq 'web.server.base.image.vhd') `
-and ($_.LibraryServer -eq 'vmm2012r2.mhc.lab') }
write-host 'The name of the acquired VHD object is '$VHD.Name
write-host 'The Library Server of the acquired VHD object is '$VHD.LibraryServer
The name of the acquired VHD object is web.server.base.image.vhd
The Library Server of the acquired VHD object is vmm2012r2.mhc.lab
PS C:\>
Completed
Ln 16 Col 9
100%

FIGURE 5-11 Windows PowerShell executes the commands to get a VHD object from a VMM library server.

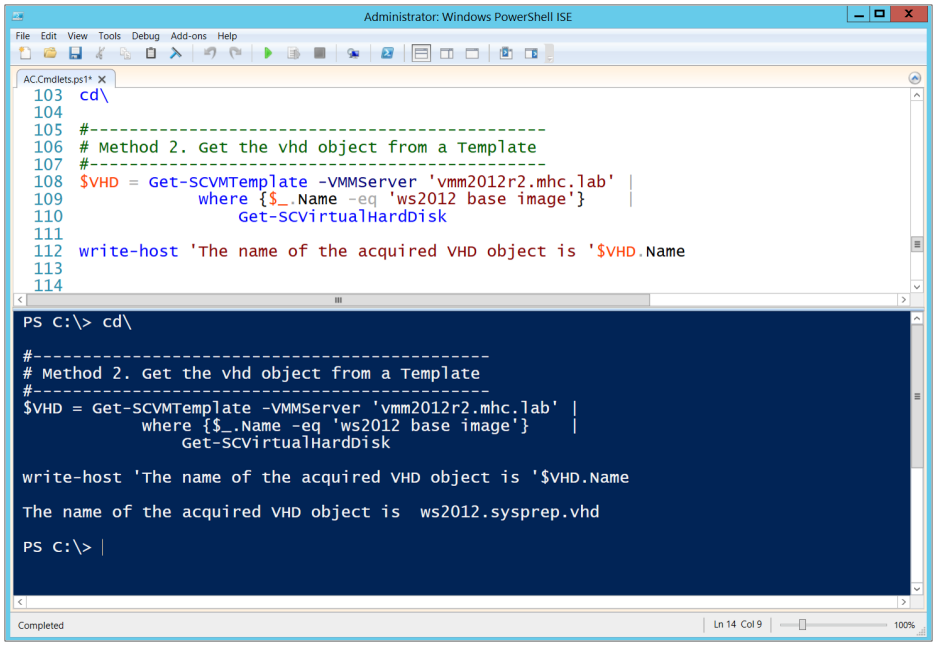

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
AC.Cmdlets.ps1*
103 cd\
104
105 #---
106 # Method 2. Get the vhd object from a Template
107 #---
108 $VHD = Get-SCVMTemplate -VMMServer 'vmm2012r2.mhc.lab' |
109 where {$_.Name -eq 'ws2012 base image'} |
110 Get-SCVirtualHardDisk
111
112 write-host 'The name of the acquired VHD object is '$VHD.Name
113
114
PS C:\> cd\
#---
Method 2. Get the vhd object from a Template
#---
$VHD = Get-SCVMTemplate -VMMServer 'vmm2012r2.mhc.lab' |
where {$_.Name -eq 'ws2012 base image'} |
Get-SCVirtualHardDisk
write-host 'The name of the acquired VHD object is '$VHD.Name
The name of the acquired VHD object is ws2012.sysprep.vhd
PS C:\>
Completed
Ln 14 Col 9
100%

FIGURE 5-12 Windows PowerShell executes the commands to get a VHD object from a template.

Finally, if you need to get a VHD object from a virtual machine, you first need to establish a connection with the associated VMM server. The returned VM object reveals its virtual disk name without the file type .vhd in the field VirtualHardDisks. So for example, if the VHD is mhc.dc10.vhd, the VirtualHardDisks field displays this as mhc.dc10. The following example shows how to acquire a VHD object from a virtual machine:

```
$VHD = Get-SCVirtualMachine -Name 'myVMdisplayName' `
    -VMMServer (Get-SCVMMServer -ComputerName 'myVMMserverFQDN') |
    Get-SCVirtualHardDisk |
    where { $_.Name -match 'myVHDfileNameWithoutDotVHDFileType' }
```

The Windows PowerShell statement in Figure 5-13 demonstrates acquiring the VHD object of a virtual machine with the display name server10 on the VMM server vmm2012r2.mhc.lab. The VHD file name is mhc.dc10.vhd, however, in the VM object the VirutalDisksFile field shows the VHD file name without the file type. This is why when filtering the results based on the VHD name, you need not include the file type and can use mhc.dc10 for name-matching instead.

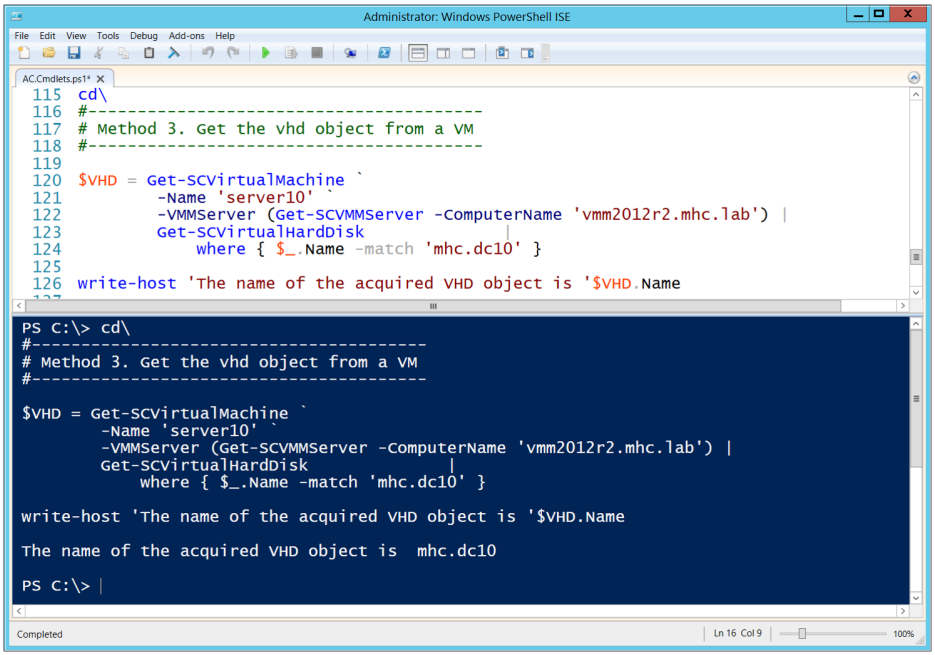

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
AC.Cmdlets.ps1*
115 cd\
116 #--
117 # Method 3. Get the vhd object from a VM
118 #--
119
120 $VHD = Get-SCVirtualMachine `
121 -Name 'server10' `
122 -VMMServer (Get-SCVMMServer -ComputerName 'vmm2012r2.mhc.lab') |
123 Get-SCVirtualHardDisk |
124 where { $_.Name -match 'mhc.dc10' }
125
126 write-host 'The name of the acquired VHD object is '$VHD.Name
PS C:\> cd\
#--
Method 3. Get the vhd object from a VM
#--
$VHD = Get-SCVirtualMachine `
-Name 'server10' `
-VMMServer (Get-SCVMMServer -ComputerName 'vmm2012r2.mhc.lab') |
Get-SCVirtualHardDisk |
where { $_.Name -match 'mhc.dc10' }
write-host 'The name of the acquired VHD object is '$VHD.Name
The name of the acquired VHD object is mhc.dc10
PS C:\>
Completed
Ln 16 Col 9
100%

FIGURE 5-13 Windows PowerShell executes the command to get a VHD object from a virtual machine.

About the authors

Yung Chou

Yung Chou is a Technology Evangelist on the Microsoft US Developer and Platform Evangelism team. Within the company, he has had opportunities serving customers in the areas of support account management, technical support, technical sales, and evangelism. Prior to Microsoft, he established capacities in system programming, application development, consulting services, and IT management. His technical focuses have been on virtualization and cloud computing with strong interests in private cloud, hybrid cloud, and emerging enterprise computing architecture. He has been a frequent speaker at Microsoft conferences, roadshows, and TechNet events.

You can find Yung online at *http://yungchou.com*.

Keith Mayer

Keith Mayer is a Senior Technical Evangelist at Microsoft focused on Windows Infrastructure, Data Center Virtualization, Systems Management, Private Cloud, and Hybrid Cloud. Keith has over 20 years of experience as a technical leader of complex IT projects, in diverse roles such as Network Engineer, IT Manager, Technical Instructor, and Consultant. He has consulted and trained thousands of IT professionals worldwide on the design and implementation of enterprise technology solutions.

Keith is currently certified on several Microsoft technologies, including Private Cloud, System Center, Hyper-V, Windows, Windows Server, SharePoint, and Exchange. He also holds other industry certifications from IBM, Cisco, VMware, Citrix, HP, CheckPoint, CompTIA, and Interwoven.

You can find Keith online at *http://KeithMayer.com*.

About the series editor

MITCH TULLOCH is a well-known expert on Windows Server administration and virtualization. He has published hundreds of articles on a wide variety of technology sites and has written or contributed to over two dozen books, including *Windows 7 Resource Kit* (Microsoft Press, 2009), for which he was lead author; *Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter* (Microsoft Press, 2010); and *Introducing Windows Server 2012* (Microsoft Press, 2012), a free ebook that has downloaded almost three-quarters of a million times.

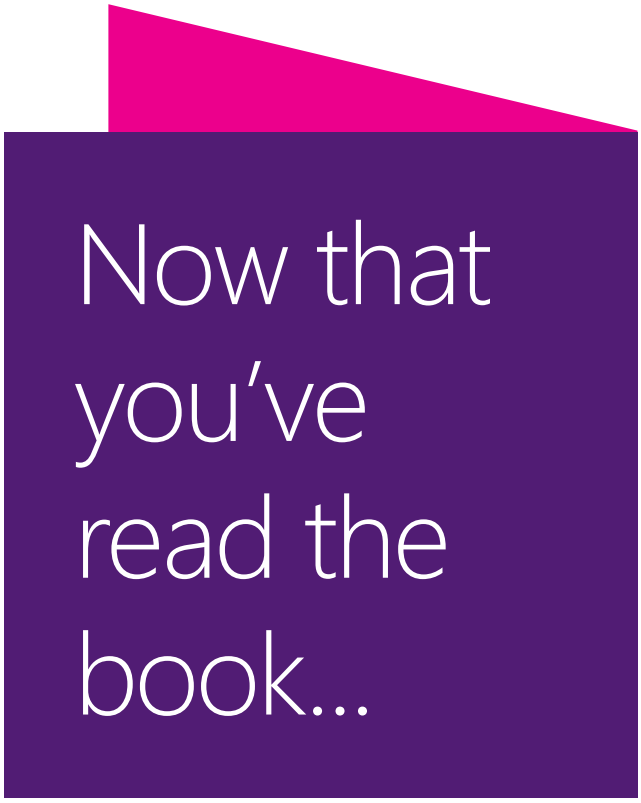
Mitch has been repeatedly awarded Most Valuable Professional (MVP) status by Microsoft for his outstanding contributions to supporting the global IT community. He is a nine-time MVP in the technology area of Windows Server Software Packaging, Deployment & Servicing. You can find his MVP Profile page at *http://mvp.microsoft.com/en-us/mvp/Mitch%20Tulloch-21182*.

Mitch is also Senior Editor of WServerNews (*http://www.wservernews.com*), a weekly newsletter focused on system administration and security issues for the Windows Server platform. With more than 100,000 IT pro subscribers worldwide, WServerNews is the largest Windows Server–focused newsletter in the world.

Mitch runs an IT content development business based in Winnipeg, Canada, that produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at *http://www.mtit.com*.

You can also follow Mitch on Twitter at *http://twitter.com/mitchtulloch* or like him on Facebook at *http://www.facebook.com/mitchtulloch*.



Tell us what you think!

Was it useful?
Did it teach you what you wanted to learn?
Was there room for improvement?

Let us know at http://aka.ms/tellpress

Your feedback goes directly to the staff at Microsoft Press,
and we read every one of your responses. Thanks in advance!

Microsoft